

# From Drafting Guideline to Error Detection: Automating Style Checking for Legislative Texts

**Stefan Höfler**

University of Zurich, Institute  
of Computational Linguistics  
Binzmühlestrasse 14  
8050 Zürich, Switzerland  
hoefler@cl.uzh.ch

**Kyoko Sugisaki**

University of Zurich, Institute  
of Computational Linguistics  
Binzmühlestrasse 14  
8050 Zürich, Switzerland  
sugisaki@cl.uzh.ch

## Abstract

This paper reports on the development of methods for the automated detection of violations of style guidelines for legislative texts, and their implementation in a prototypical tool. To this aim, the approach of error modelling employed in automated style checkers for technical writing is enhanced to meet the requirements of legislative editing. The paper identifies and discusses the two main sets of challenges that have to be tackled in this process: (i) the provision of domain-specific NLP methods for legislative drafts, and (ii) the concretisation of guidelines for legislative drafting so that they can be assessed by machine. The project focuses on German-language legislative drafting in Switzerland.

## 1 Introduction

This paper reports on work in progress that is aimed at providing domain-specific automated style checking to support German-language legislative editing in the Swiss federal administration. In the federal administration of the Swiss Confederation, drafts of new acts and ordinances go through several editorial cycles. In a majority of cases, they are originally written by civil servants in one of the federal offices concerned, and then reviewed and edited both by legal experts (at the Federal Office of Justice) and language experts (at the Federal Chancellery). While the former ensure that the drafts meet all relevant legal requirements, the latter are concerned with the formal and linguistic quality of the texts. To help this task, the authorities have drawn up style guidelines specifically geared towards Swiss legislative texts (Bundeskanzlei, 2003; Bundesamt für Justiz, 2007).

Style guidelines for laws (and other types of legal texts) may serve three main purposes: (i) improving the understandability of the texts (Lerch, 2004; Wydick, 2005; Mindlin, 2005; Butt and Castle, 2006; Eichhoff-Cyrus and Antos, 2008), (ii) enforcing their consistency with related texts, and (iii) facilitating their translatability into other languages. These aims are shared with writing guidelines developed for controlled languages in the domain of technical documentation (Lehrndorfer, 1996; Reuther, 2003; Muegge, 2007).

The problem is that the manual assessment of draft laws for their compliance with all relevant style guidelines is time-consuming and easily inconsistent due to the number of authors and editors involved in the drafting process. The aim of the work presented in this paper is to facilitate this process by providing methods for a consistent automatic identification of some specific guideline violations.

The remainder of the paper is organised as follows. We first delineate the aim and scope of the project presented in the paper (section 2) and the approach we are pursuing (section 3). In the main part of the paper, we then identify and discuss the two main challenges that have to be tackled: the technical challenge of providing NLP methods for legislative drafts (section 4) and the linguistic challenge of concretising the existing drafting guidelines for legislative texts (section 5).

## 2 Aim and Scope

The aim of the project to be presented in this paper is to develop methods of automated style checking specifically geared towards legislative editing, and to implement these methods in a prototypical tool (cf. sections 3 and 4). We work towards automat-

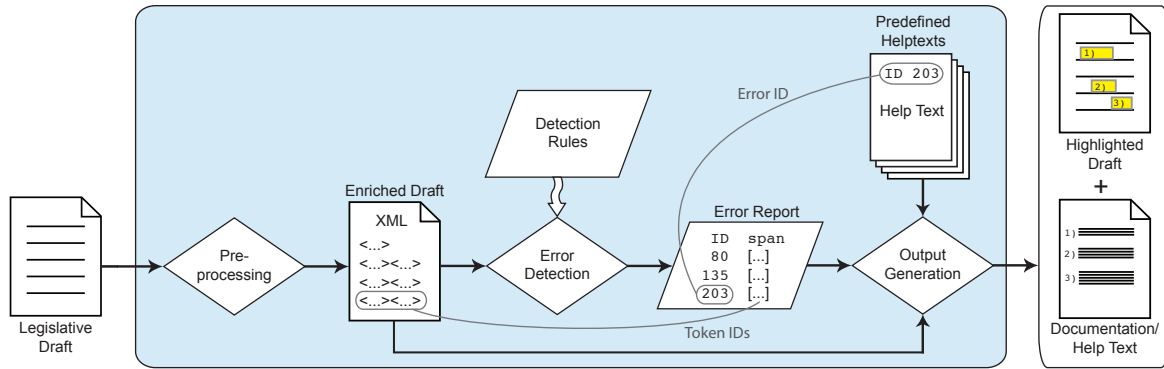


Figure 1: Architecture of the style checking system.

ically detecting violations of existing guidelines, and where these guidelines are very abstract, we concretise them so that they become detectable by machine (cf. section 5). However, it is explicitly not the goal of our project to propose novel style rules.

We have adopted a broad conception of “style checking” that is roughly equivalent to how the term, and its variant “controlled language checking,” have been used in the context of technical writing (Geldbach, 2009). It comprises the assessment of various aspects of text composition controlled by specific writing guidelines (typographical conventions, lexical preferences, syntax-related recommendations, constraints on discourse and document structure), but it does not include the evaluation of spelling and grammar.

While our project focuses on style checking for German-language Swiss federal laws (the federal constitution, acts of parliament, ordinances, federal decrees, cantonal constitutions), we believe that the challenges arising from the task are independent of the chosen language and legislative system but pertain to the domain in general.

### 3 Approach

The most important innovative contribution of our project is the enhancement of the method of error modelling to meet the requirements of legislative editing. Error modelling means that texts are searched for specific features that indicate a style guideline violation: the forms of specific “errors” are thus anticipated and modelled.

The method of error modelling has mainly been developed for automated style checking in the domain of technical writing. Companies often con-

trol the language used in their technical documentation in order to improve the understandability, readability and translatability of these texts. Controlled language checkers are tools that evaluate input texts for compliance with such style guidelines set up by a company.<sup>1</sup>

State-of-the-art controlled language checkers work along the following lines. In a pre-processing step, they first perform an automatic analysis of the input text (tokenisation, text segmentation, morphological analysis, part-of-speech tagging, parsing) and enrich it with the respective structural and linguistic information. They then apply a number of pre-defined rules that model potential “errors” (i.e. violations of individual style guidelines) and aim at detecting them in the analysed text. Most checkers give their users the option to choose which rules the input text is to be checked for. Once a violation of the company’s style guidelines has been detected, the respective passage is highlighted and an appropriate help text is made available to the user (e.g. as a comment in the original document or in an extra document generated by the system). The system we are working on is constructed along the same lines; its architecture is outlined in Fig. 1.

Transferring the described method to the domain of legislative editing has posed challenges to both pre-processing and error modelling. The peculiarities of legal language and legislative texts have necessitated a range of adaptations in the NLP procedures devised, and the guidelines for legislative drafting have required highly domain-specific

<sup>1</sup>Examples of well-developed commercial tools that offer such style checking for technical texts are acrolinx IQ by Acrolinx and CLAT by IAI.

error modelling, which needed to be backed up by substantial linguistic research. We will detail these two sets of challenges in the following two sections.

## 4 Pre-Processing

### 4.1 Tokenisation

The legislative drafters and editors we are targeting exclusively work with MS Word documents. Drafters compose the texts in Word, and legislative editors use the commenting function of Word to add their suggestions and corrections to the texts they receive. We make use of the XML representation (WordML) underlying these documents. In a first step, we tokenise the text contained therein and assign each token an ID directly in the WordML structure. We then extract the text material (including the token IDs and some formatting information that proves useful in the processing steps to follow) for further processing. The token IDs are used again at the end of the style checking process when discovered styleguide violations are highlighted by inserting a Word comment at the respective position in the WordML representation of the original document. The output of our style checker is thus equivalent to how legislative editors make their annotations to the drafts – a fact that proves essential with regard to the tool being accepted by its target users.

### 4.2 Text Segmentation

After tokenisation, the input text is then segmented into its structural units. Legislative texts exhibit a sophisticated domain-specific structure. Our text segmentation tool detects the boundaries of chapters, sections, articles, paragraphs, sentences and enumeration elements, and marks them by adding corresponding XML tags to the text.

There are three reasons why text segmentation is crucial to our endeavour:

1. Proper text segmentation ensures that only relevant token spans are passed on to further processing routines (e.g. sentences contained in articles must to be passed on to the parser, whereas article numbers or section headings must not).
2. Most structural units are themselves the object of style rules (e.g. “sections should not contain more than twelve articles, articles

should not contain more than three paragraphs and paragraphs should not contain more than one sentence”). The successful detection of violations of such rules depends on the correct delimitation of the respective structural units in the text.

3. Certain structural units constitute the context for other style rules (e.g. “the sentence right before the first element of an enumeration has to end in a colon”; “the antecedent of a pronoun must be within the same article”). Here too, correct text segmentation constitutes the prerequisite for an automated assessment of the respective style rules.

We have devised a line-based pattern-matching algorithm with look-around to detect the boundaries of the structural units of legislative drafts (Höfler and Piotrowski, 2011). The algorithm also exploits formatting information extracted together with the text from the Word documents. However, not all formatting information has proven equally reliable: as the Word documents in which the drafts are composed do only make use of style environments to a very limited extent, formatting errors are relatively frequent. Font properties such as italics or bold face, or the use of list environments are frequently erroneous and can thus not be exploited for the purpose of delimiting text segments; headers and newline information, on the other hand, have proven relatively reliable.

Figure 2 illustrates the annotation that our tool yields for the excerpt shown in the following example:

(1) *Art. 14 Amtsenthebung*<sup>2</sup>

Die Wahlbehörde kann eine Richterin oder einen Richter vor Ablauf der Amtsdauer des Amtes entheben, wenn diese oder dieser:

- a. vorsätzlich oder grobfahrlässig Amtspflichten schwer verletzt hat; oder
- b. die Fähigkeit, das Amt auszuüben, auf Dauer verloren hat.

*Art. 14 Removal from office*

The electoral authorities may remove a judge from office before he or she has completed his or her term where he or she:

<sup>2</sup>Patentgerichtsgesetz (Patent Court Act), SR 173.41; for the convenience of readers, examples are also rendered in the (non-authoritative) English version published at <http://www.admin.ch/ch/e/rs/rs.html>.

```

<article>
  <article_head>
    <article_type>Art.</article_type>
    <article_nr>14</article_nr>
    <article_header>Amtsenthebung</article_header>
  </article_head>
  <article_body>
    <paragraph>
      <sentence>
        Die Wahlbehörde kann eine Richterin oder einen Richter vor Ablauf der Amtsdauer
        des Amtes entheben, wenn diese oder dieser:
        <enumeration>
          <enumeration_element>
            <element_nr type="letter">a.</element_nr>
            <element_text>
              vorsätzlich oder grobfahrlässig Amtspflichten schwer verletzt hat;
            oder
          </element_text>
        </enumeration_element>
        <enumeration_element>
          <element_nr type="letter">b.</element_nr>
          <element_text>
            die Fähigkeit, das Amt auszuüben, auf Dauer verloren hat.
          </element_text>
        </enumeration_element>
      </enumeration>
    </sentence>
  </paragraph>
</article_body>
</article>

```

Figure 2: Illustration of the text segmentation provided by the tool. Excerpt: Article 14 of the Patent Court Act. (Token delimiters and any other tags not related to text segmentation have been omitted in the example.)

- a. wilfully or through gross negligence commits serious breaches of his or her official duties; or
- b. has permanently lost the ability to perform his or her official duties.

As our methods must be robust in the face of input texts that are potentially erroneous, the text segmentation provided by our tool does not amount to a complete document parsing; our text segmentation routine rather performs a document chunking by trying to detect as many structural units as possible.

Another challenge that arises from the fact that the input texts may be erroneous is that features whose absence we later need to mark as an error cannot be exploited for the purpose of detecting the boundaries of the respective contextual unit. A colon, for instance, cannot be used as an indicator for the beginning of an enumeration since we must later be able to search for enumerations that are not preceded by a sentence ending in a colon as this constitutes a violation of the respective style rule. Had the colon been used as an indicator for the detection of enumeration boundaries, only enumerations preceded by a colon would have been marked

as such in the first place. The development of adequate pre-processing methods constantly faces such dilemmas. It is thus necessary to always anticipate the specific guideline violations that one later wants to detect on the basis of the information added by any individual pre-processing routine.

Special challenges also arise with regard to the task of sentence boundary detection. Legislative texts contain special syntactic structures that off-the-shelf tools cannot process and that therefore need special treatment. Example (1) showed a sentence that runs throughout a whole enumeration; colon and semicolons do not mark sentence boundaries in this case. To complicate matters even further, parenthetical sentences may be inserted behind individual enumeration items, as shown in example (2).

(2) *Art. 59 Abschirmung*<sup>3</sup>

<sup>1</sup> Der Raum oder Bereich, in dem stationäre Anlagen oder radioaktive Strahlenquellen betrieben oder gelagert werden, ist so zu

<sup>3</sup>Strahlenschutzverordnung (Radiological Protection Ordinance), SR 814.50; emphasis added.

konzipieren oder abzuschirmen, dass unter Berücksichtigung der Betriebsfrequenz:

- a. an Orten, die zwar innerhalb des Betriebsareals, aber ausserhalb von kontrollierten Zonen liegen und an denen sich nichtberuflich strahlenexponierte Personen aufhalten können, die Ortsdosis 0,02 mSv pro Woche nicht übersteigt. **Dieser Wert kann an Orten, wo sich Personen nicht dauernd aufhalten, bis zum Fünffachen überschritten werden;**
- b. an Orten ausserhalb des Betriebsareals die Immissionsgrenzwerte nach Artikel 102 nicht überschritten werden.

<sup>2</sup> [...]

#### Art. 59 Shielding

<sup>1</sup> The room or area in which stationary radiation generators or radioactive sources are operated or stored shall be designed and shielded in such a way that, taking into account the frequency of use:

- a. in places situated within the premises but outside controlled areas, where non-occupationally exposed persons may be present, the local dose does not exceed 0.02 mSv per week. **In places where people are not continuously present, this value may be exceeded by up to a factor of five;**
- b. in places outside the premises, the off-site limits specified in Article 102 are not exceeded.

<sup>2</sup> [...]

In this example, a parenthetical sentence (marked in bold face) has been inserted at the end of the first enumeration item. A full stop has been put where the main sentence is interrupted, whereas the inserted sentence is ended with a semicolon to indicate that after it, the main sentence is continued. The recognition of sentential insertions as the one shown in (2) is important for two reasons: (i) sentential parentheses are themselves the object of style rules (in general, they are to be avoided) and should thus be marked by a style checker, and (ii) a successful parsing of the texts depends on a proper recognition of the sentence boundaries. As

off-the-shelf tools cannot cope with such domain-specific structures, we have had to devise highly specialised algorithms for sentence boundary detection in our texts.

### 4.3 Linguistic Analysis

Following text segmentation, we perform a linguistic analysis of the input text which consists of three components: part-of-speech tagging, lemmatisation and chunking/parsing. The information added by these pre-processing steps is later used in the detection of violations of style rules that pertain to the use of specific terms (e.g. “the modal *sollen* ‘should’ is to be avoided”), syntactic constructions (e.g. “complex participial constructions preceding a noun should be avoided”) or combinations thereof (e.g. “obligations where the subject is an authority must be put as assertions and not contain a modal verb”).

For the tasks of part-of-speech tagging and lemmatisation, we employ TreeTagger (Schmid, 1994). We have adapted TreeTagger to the peculiarities of Swiss legislative language. Domain-specific token types are pre-tagged in a special routine to avoid erroneous part-of-speech analyses. An example of a type of tokens that needs pre-tagging are domain-specific cardinal numbers: i.e. cardinal numbers augmented with letters (*Article 2a*) or with Latin ordinals (*Paragraph 4bis*) as well as ranges of such cardinal numbers (*Articles 3c–6*). Furthermore, TreeTagger’s recognition of sentence boundaries is overwritten by the output of our text segmentation routine. We have also augmented TreeTagger’s domain-general list of abbreviations with a list of domain-specific abbreviations and acronyms provided by the Swiss Federal Chancellery. The lemmatisation provided by TreeTagger usually does not recognise complex compound nouns (e.g. *Güterverkehrsverlagerung* ‘freight traffic transfer’); such compound nouns are frequent in legislative texts (Nussbaumer, 2009). To solve the problem, we combine the output of TreeTagger’s part-of-speech tagging with the lemma information delivered by the morphology analysis tool GERTWOL (Haapalainen and Majorin, 1995).

Some detection tasks (e.g. the detection of legal definitions discussed in section 4.4 below) additionally require chunking or even parsing. For chunking, we also employ TreeTagger; for parsing, we have begun to adapt ParZu to legislative language, a robust state-of-art dependency parser

(Sennrich et al., 2009). Like most off-the-shelf parsers, ParZu was trained on a corpus of newspaper articles. As a consequence, it struggles with analysing constructions that are rare in that domain but frequent in legislative texts, such as complex coordinations of prepositional phrases and PP-attachment chains (Venturi, 2008), parentheses (as illustrated in example 2 above) or subject clauses (as shown in example 3 below).

- (3) *Art. 17 Rechtfertigender Notstand*<sup>4</sup>  
**Wer eine mit Strafe bedrohte Tat begeht, um ein eigenes oder das Rechtsgut einer anderen Person aus einer unmittelbaren, nicht anders abwendbaren Gefahr zu retten, handelt rechtmässig, wenn er dadurch höherwertige Interessen wahr.**

*Art. 17 Legitimate act in a situation of necessity*

**Whoever carries out an act that carries a criminal penalty in order to save a legal interest of his own or of another from immediate and not otherwise avertable danger, acts lawfully if by doing so he safeguards interests of higher value.**

As the adaptation of ParZu to legislative texts is still in its early stages, we cannot yet provide an assessment of how useful the output of the parser, once properly modified, will be to our task.

#### 4.4 Context Recognition

The annotations that the pre-processing routines discussed so far add to the text serve as the basis for the automatic recognition of domain-specific contexts. Style rules for legislative drafting often only apply to special contexts within a law. An example is the rule pertaining to the use of the modal *sollen* ('should'). The drafting guidelines forbid the use of this modal *except* in statements of purpose. Statements of purpose thus constitute a special context inside which the detection of an instance of *sollen* is not to trigger an error message. Other examples of contexts in which special style rules apply are transitional provisions (*Übergangsbestimmungen*), repeals and amendments of current legislation (*Aufhebungen und Änderungen bisherigen Rechts*), definitions of the

<sup>4</sup>Strafgesetzbuch (Criminal Code), SR 311.0; emphasis added.

subject of a law (*Gegenstandsbestimmungen*), definitions of the scope of a law (*Geltungsbereichsbestimmungen*), definitions of terms (*Begriffsbestimmungen*), as well as preambles (*Präambeln*) and commencement clauses (*Ingresse*).

A number of these contexts can be identified automatically by assessing an article's position in the text and certain keywords contained in its header. A statements of purpose, for instance, is usually the first article of a law, and its header usually contains the words *Zweck* ('purpose') or *Ziel* ('aim'). Similar rules can be applied to recognise transitional provisions, repeals and amendments of current legislation, and definitions of the subject and the scope of a law.

Other contexts have to be detected at the sentential level. Definitions of terms, for instance, do not only occur as separate articles at the beginning of a law; they can also appear in the form of individual sentences throughout the text. As there is a whole range of style rules pertaining to legal definitions (e.g. "a term must only be defined if it occurs at least three times in the text"; "a term must only be defined once within the same text"; "a term must not be defined by itself"), the detection of this particular context (and its components: the term and the actual definition) is crucial to a style checker for legislative texts.<sup>5</sup>

To identify legal definitions in the text, we have begun to adopt strategies developed in the context of legal information retrieval: Walter and Pinkal (2009) and de Maat and Winkels (2010), for instance, show that definitions in German court decisions and in Dutch laws respectively can be detected by searching for combinations of key words and sentence patterns typically used in these domain-specific contexts. In Höfler et al. (2011) we have argued that this approach is also feasible with regard to Swiss legislative texts: our pilot study has shown that a substantial number of legal definitions can be detected even without resorting to syntactic analyses, merely by searching for typical string patterns such as '*X im Sinne dieser Verordnung ist/sind Y*' ('X in the sense of this ordinance is/are Y'). We are currently working towards refining and extending the detection of legal definitions by including additional syntactic information yielded by the processes of chunking and parsing into the search patterns.

<sup>5</sup>Further rules for the use of legal definitions in Swiss law texts are provided by Bratschi (2009).

Once the legal definitions occurring in a draft have been marked, the aforementioned style rules can be checked automatically (e.g. by searching the text for terms that are defined in a definition but occur less than three times in the remainder of the text; by checking if there are any two legal definitions that define the same term; by assessing if there are definitions where the defined term also occurs in the actual definition).

After having outlined some of the main challenges that the peculiarities of legal language and legislative texts pose to the various pre-processing tasks, we now turn to the process of error modelling, i.e. the effort of transferring the guidelines for legislative drafting into concrete error detection mechanisms operating on the pre-processed texts.

## 5 Error Modelling

### 5.1 Sources

The first step towards error modelling consists in collecting the set of style rules that shall be applied to the input texts. The main source that we use for this purpose are the compilations of drafting guidelines published by the Swiss Federal Administration (Bundeskanzlei, 2003; Bundesamt für Justiz, 2007). However, especially when it comes to linguistic issues, these two documents do not claim to provide an exhaustive set of writing rules. Much more so than the writing rules that are put in place in the domain of technical documentation, the rules used in legislative drafting are based on historically grown conventions, and there may well be conventions beyond what is explicitly written down in the Federal Administration's official drafting guidelines.

Consequently, we have also been collecting rule material from three additional sources. A first complementary source are the various drafting guidelines issued by cantonal governments (Regierungsrat des Kantons Zürich, 2005; Regierungsrat des Kantons Bern, 2000) and, to a lesser extent, the drafting guidelines of the other German-speaking countries (Bundesministerium für Justiz, 2008; Bundeskanzleramt, 1990; Rechtsdienst der Regierung, 1990) and the European Union (Europäische Kommission, 2003). A second source are academic papers dealing with specific issues of legislative drafting, such as Eisenberg (2007), Bratschi (2009).

Finally, legislative editors themselves constitute an invaluable source of expert knowledge. In order to learn of their unwritten codes of practice, we have established a regular exchange with the Central Language Services of the Swiss Federal Chancellery. Including the editors in the process is likely to prove essential for the acceptability of the methods that we develop.

### 5.2 Concretisation and Formalisation

The next error modelling step consists in concretising and formalising the collected rules so that specific algorithms can be developed to search for violations of the rules in the pre-processed texts. Depending on the level of abstraction of a rule, this task is relatively straight-forward or it requires more extensive preliminary research:

**Concrete Rules** A number of rules for legislative drafting define concrete constraints and can thus be directly translated into detection rules. Examples of such concrete rules are rules that prohibit the use of specific abbreviations (e.g. *bzw.* 'respectively'; *z.B.* 'e.g.'; *d.h.* 'i.e.') and of certain terms and phrases (e.g. *grundsätzlich* 'in principle'; *in der Regel* 'as a general rule'). In such cases, error detection simply consists in searching for the respective items in the input text.

Some rules first need to be spelled out but can then also be formalised more or less directly: the rule stating that units of measurement must always be written out rather than abbreviated, for instance, requires that a list of such abbreviations of measuring units (e.g. *m* for meter, *kg* for kilogram, *%* for percent) is compiled whose entries can then be searched for in the text.

The formalisation of some other rules is somewhat more complicated but can still be derived more or less directly. The error detection strategies for these rules include accessing tags that were added during pre-processing or evaluating the environment of a potential error. For example, the rule stating that sentences introducing an enumeration must end in a colon can be checked by searching the text for <enumeration> tags that are *not* preceded by a colon; violations of the rule stating that an article must not contain more than three paragraphs can be detected by counting for each <article\_body> environment, the number of <paragraph> elements it contains.

**Abstract Rules** However, guidelines for legislative drafting frequently contain rules that define relatively abstract constraints. In order to be able to detect violations of such constraints, a linguistic concretisation of the rules is required.

An example is the oft-cited rule that a sentence should only convey one statement or proposition (Bundesamt für Justiz, 2007, p. 358). The error modelling for this rule is not straightforward: it is neither clear what counts as a statement in the context of a legislative text, nor is it obvious what forms sentences violating this rule exhibit. Linguistic indicators for the presence of a multi-propositional sentence first need to be determined in in-depth analyses of legislative language. In Höfler (2011), we name a number of such indicators: among other things, sentence coordination, relative clauses introduced by the adverb *wobei* ('whereby'), and certain prepositions (e.g. *vorbehältlich* 'subject to' or *mit Ausnahme von* 'with the exception of') can be signs that a sentence contains more than one statement.

Even drafting rules that look fairly specific at first glance may turn out to be in need of further linguistic concretisation. An example is the rule that states that in an enumeration, words that are shared between all enumeration elements should be bracketed out into the introductory sentence of the enumeration. If, for instance, each element of an enumeration starts with the preposition *für* ('for'), then that preposition belongs in the introductory sentence. The rule seems straight enough, but in reality, the situation is somewhat more complicated. Example (4) shows a case where a word that occurs at the beginning of all elements of an enumeration (the definite article *die* 'the') cannot be bracketed out into the introductory sentence:

(4) *Art. 140 Obligatorisches Referendum*<sup>6</sup>  
[...]

<sup>2</sup> Dem Volk werden zur Abstimmung unterbreitet:

- a. **die** Volksinitiativen auf Totalrevision der Bundesverfassung;
- b. **die** Volksinitiativen auf Teilrevision der Bundesverfassung in der Form der allgemeinen Anregung, die von der Bundesversammlung abgelehnt worden sind;

<sup>6</sup>Bundesverfassung (Federal Constitution), SR 101; emphasis added.

- c. **die** Frage, ob eine Totalrevision der Bundesverfassung durchzuführen ist, bei Uneinigkeit der beiden Räte.

*Art. 140 Mandatory referendum*  
[...]

<sup>2</sup> The following shall be submitted to a vote of the People:

- a. **the** popular initiatives for a complete revision of the Federal Constitution;
- b. **the** popular initiatives for a partial revision of the Federal Constitution in the form of a general proposal that have been rejected by the Federal Assembly;
- c. **the** question of whether a complete revision of the Federal Constitution should be carried out, in the event that there is disagreement between the two Councils.

Even if one ignores the fact that the definite article in letters *a* and *b* is in fact not the same as the one in letter *c* (the former being plural, the latter singular), it is quite apparent that articles cannot be extracted from the elements of an enumeration without the nouns they specify. Even the seemingly simple rule in question is thus in need of a more linguistically informed concretisation before it can be effectively checked by machine.

The examples illustrate that style guidelines for legislative writing are often kept at a level of abstraction that necessitates concretisations if one is to detect violations of the respective rules automatically. Besides the development of domain-specific pre-processing algorithms, the extensive and highly specialised linguistic research required for such concretisations constitutes the main task being tackled in this project.

**Conflicting Rules** A further challenge to error modelling arises from the fact that a large proportion of drafting guidelines for legislative texts do not constitute absolute constraints but rather have the status of general writing principles and rules of thumb. This fact has to be reflected in the feedback messages that the system gives to its users: what the tool detects are often not "errors" in the proper sense of the word but merely passages that the author or editor may want to reconsider.

The fact that many style rules only define soft constraints also means that there may be conflicting rules. Consider, for instance, sentence (5):



(5) *Art. 36 Ersatzfreiheitsstrafe*<sup>7</sup>

[...]

<sup>5</sup> Soweit der Verurteilte die Geldstrafe trotz verlängerter Zahlungsfrist oder herabgesetztem Tagessatz nicht bezahlt oder die gemeinnützige Arbeit trotz Mahnung nicht leistet, wird die Ersatzfreiheitsstrafe vollzogen.

*Art. 36 Alternative custodial sentence*

[...]

<sup>5</sup> As far as the offender fails to pay the monetary penalty despite being granted an extended deadline for payment or a reduced daily penalty unit or fails to perform the community service despite being warned of the consequences, the alternative custodial sentence is executed.

On the one hand, this sentence must be considered a violation of the style rule that states that the main verb of a sentence (here *execute*) should be introduced as early as possible (Regierungsrat des Kantons Zürich, 2005, p. 73). On the other hand, if the sentence was re-arranged in compliance with this rule – by switching the order of the main clause and the subsidiary clause – it would violate the rule stating that information is to be presented in temporal and causal order (Bundesamt für Justiz, 2007, p. 354). This latter rule entails that the condition precedes its consequence.

To be able to deal with such conflicting constraints, error detection strategies have to be assigned weights. However, one and the same rule may have different weights under different circumstances. In conditional sentences like the one shown above, the causality principle obviously weighs more than the rule that the main verb must be introduced early in the sentence. Such context-dependent rankings for individual style rules have to be inferred and corroborated by tailor-made corpus-linguistic studies.

### 5.3 Testing and Evaluation

The number of drafts available to us is very limited – too limited to be used to test and refine the error models we develop. However, due to the complexity of the drafting process (multiple authors and editors, political intervention), laws that

have already come into force still exhibit violations of specific style rules. We therefore resort to such already published laws to test and refine the error models we develop. To this aim, we have built a large corpus of legislative texts automatically annotated by the pre-processing routines we have described earlier in the paper (Höfler and Piotrowski, 2011). The corpus contains the entire current federal legislation of Switzerland, i.e. the federal constitution, all cantonal constitutions, all federal acts and ordinances, federal decrees and treaties between the Confederation and individual cantons and municipalities. It allows us to try out and evaluate novel error detection strategies by assessing the number and types of true and false positives returned.

## 6 Conclusion

In this paper, we have discussed the development of methods for the automated detection of violations of domain-specific style guidelines for legislative texts, and their implementation in a prototypical tool. We have illustrated how the approach of error modelling employed in automated style checkers for technical writing can be enhanced to meet the requirements of legislative editing. Two main sets of challenges are tackled in this process. First, domain-specific NLP methods for legislative drafts have to be provided. Without extensive adaptations, off-the-shelf NLP tools that have been trained on corpora of newspaper articles are not adequately equipped to deal with the peculiarities of legal language and legislative texts. Second, the error modelling for a large number of drafting guidelines requires a concretisation step before automated error detection strategies can be put in place. The substantial linguistic research that such concretisations require constitutes a core task to be carried out in the development of a style checker for legislative texts.

## Acknowledgments

The project is funded under SNSF grant 134701. The authors wish to thank the Central Language Services of the Swiss Federal Chancellery for their continued advice and support.

## References

Rebekka Bratschi. 2009. “Frau im Sinne dieser Badeordnung ist auch der Bademeister.” Legaldefinition-

<sup>7</sup>Strafgesetzbuch (Criminal Code), SR 311.0

- nen aus redaktioneller Sicht. *LeGes*, 20(2):191–213.
- Bundesamt für Justiz, editor. 2007. *Gesetzgebungsleitfaden: Leitfaden für die Ausarbeitung von Erlassen des Bundes*. Bern, 3. edition.
- Bundeskanzlei, editor. 2003. *Gesetzestechische Richtlinien*. Bern.
- Bundeskanzleramt, editor. 1990. *Handbuch der Rechtsetzungstechnik, Teil 1: Legistische Leitlinien*. Wien.
- Bundesministerium für Justiz, editor. 2008. *Handbuch der Rechtsförmlichkeit, Empfehlungen zur Gestaltung von Gesetzen und Rechtsverordnungen*. Bundesanzeiger Verlag, Köln.
- Peter Butt and Richard Castle. 2006. *Modern Legal Drafting*. Cambridge University Press, Cambridge, UK, 2nd edition.
- Emile de Maat and Radboud Winkels. 2010. Automated classification of norms in sources of law. In *Semantic Processing of Legal Texts*. Springer, Berlin.
- Karin M. Eichhoff-Cyrus and Gerd Antos, editors. 2008. *Verständlichkeit als Bürgerrecht? Die Rechts- und Verwaltungssprache in der öffentlichen Diskussion*. Duden, Mannheim, Germany.
- Peter Eisenberg. 2007. Die Grammatik der Gesetzesprache: Was ist eine Verbesserung? In Andreas Lötscher and Markus Nussbaumer, editors, *Denken wie ein Philosoph und schreiben wie ein Bauer*, pages 105–122. Schulthess, Zürich.
- Europäische Kommission, editor. 2003. *Gemeinsamer Leitfaden des Europäischen Parlaments, des Rates und der Kommission für Personen, die in den Gemeinschaftsorganen an der Abfassung von Rechtstexten mitwirken*. Amt für Veröffentlichungen der Europäischen Gemeinschaften, Luxemburg.
- Stephanie Geldbach. 2009. Neue Werkzeuge zur Autorenunterstützung. *MDÜ*, 4:10–19.
- Mariikka Haapalainen and Ari Majorin. 1995. GER-TWOL und morphologische Desambiguierung für das Deutsche. In *Proceedings of the 10th Nordic Conference of Computational Linguistics*. University of Helsinki, Department of General Linguistics.
- Stefan Höfler and Michael Piotrowski. 2011. Building corpora for the philological study of Swiss legal texts. *Journal for Language Technology and Computational Linguistics (JLCL)*, 26(2):77–90.
- Stefan Höfler, Alexandra Bünzli, and Kyoko Sugisaki. 2011. Detecting legal definitions for automated style checking in draft laws. Technical Report CL-2011.01, University of Zurich, Institute of Computational Linguistics, Zürich.
- Stefan Höfler. 2011. “Ein Satz – eine Aussage.” Multipropositionale Rechtssätze an der Sprache erkennen. *LeGes*, 22(2):259–279.
- Anne Lehrndorfer. 1996. *Kontrolliertes Deutsch: Linguistische und sprachpsychologische Leitlinien für eine (maschinell) kontrollierte Sprache in der Technischen Dokumentation*. Günter Narr, Tübingen.
- Kent D. Lerch, editor. 2004. *Recht verstehen. Verständlichkeit, Missverständlichkeit und Unverständlichkeit von Recht*. de Gruyter, Berlin.
- Maria Mindlin. 2005. Is plain language better? A comparative readability study of plain language court forms. *Scribes Journal of Legal Writing*, 10.
- Uwe Muegge. 2007. Controlled language: The next big thing in translation? *ClientSide News Magazine*, 7(7):21–24.
- Markus Nussbaumer. 2009. Rhetorisch-stilistische Eigenschaften der Sprache des Rechtswesens. In Ulla Fix, Andreas Gardt, and Joachim Knappe, editors, *Rhetorik und Stilistik/Rhetoric and Stylistics*, Handbooks of Linguistics and Communication Science, pages 2132–2150. de Gruyter, New York/Berlin.
- Rechtsdienst der Regierung, editor. 1990. *Richtlinien der Regierung des Fürstentums Liechtenstein über die Grundsätze der Rechtsetzung (Legistische Richtlinien)*. Vaduz.
- Regierungsrat des Kantons Bern, editor. 2000. *Rechtsetzungsrichtlinien des Kantons Bern*. Bern.
- Regierungsrat des Kantons Zürich, editor. 2005. *Richtlinien der Rechtsetzung*. Zürich.
- Ursula Reuther. 2003. Two in one – can it work? Readability and translatability by means of controlled language. In *Proceedings of EAMT-CLAW 2003*.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.
- Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. A new hybrid dependency parser for German. In *Proceedings of the GSCL Conference 2009*, pages 115–124, Tübingen.
- Giulia Venturi. 2008. Parsing legal texts: A contrastive study with a view to knowledge management applications. In *Proceedings of the LREC 2008 Workshop on Semantic Processing of Legal Texts*, pages 1–10, Marakesh.
- Stephan Walter and Manfred Pinkal. 2009. Definitions in court decisions: Automatic extraction and ontology acquisition. In Joost Breuker, Pompeu Casanovas, Michel Klein, and Enrico Francesconi, editors, *Law, Ontologies and the Semantic Web*. IOS Press, Amsterdam.
- Richard C. Wydick. 2005. *Plain English for Lawyers*. Carolina Academic Press, 5th edition.