A First Effort to Create a Categorization Scheme for Analyzing a Handbook of Swedish Writing Rules

Jody Foo
Linköping University
Linköping, Sweden
jody.foo@liu.se

Abstract

Today, spelling and grammar checkers are integrated into modern word processing environments. In some contexts of writing however, these components are not sufficient. In companies that write technical documentation, or when writing a research paper for a specific scientific community, style guides that can only be found in handbooks need to be followed. If such rules are to be implemented in a language-checking framework, they need to be analyzed to identify the requirements on the framework. A categorization scheme for such analysis does not seem to exist, hence the contribution of this paper – a first attempt at a scheme for classifying style guide rules for future implementation.

1 Background

There are many kinds of errors that can be made in written texts and there have been many attempts to automate the detection and correction of such errors. The most common kind of language checkers are spelling and grammar checkers for general language, e.g., those included in many word processing applications today. However, there are cases where a more restricted or specialized written language is needed, e.g., in companies that produce technical documentation (Almqvist & Hein) written in Simplified Technical English[1] (or similar controlled language), and when writing research publications that have to follow certain style guides such as APA style[2]. Such guidelines and rules for writing mostly exist in the form of written handbooks. Handbooks are however not that practical during the actual writing. The writer needs to keep an active knowledge of the guidelines and rules to be able to use them.

Analysis of writing rules contained in handbooks is needed to develop a framework capable of incorporating such guidelines.

To the author's knowledge, no research has been conducted concerning the problem of transforming written rules from style guides and handbooks into rules used in a language checker.

In this paper, we discuss a categorization scheme for analysis of such written rules for future implementation.

2 Rule/Error Types

There are many kinds of errors that can be found in written text. Below, a top-level categorization of these error types is presented. It should be noted that a classification scheme applied to errors is also applicable to rules. For example a spelling error is the result of breaking a spelling rule, e.g., using a word not contained by the dictionary. Categorization schemes exist for spelling errors and grammar errors but these are often considered separate problems from an implementation point of view (Domeij, Knutsson, Carlberger, & Kann, 1999). The scheme presented below combines some previous efforts but also contributes by taking into account the experiences from the analysis described in section 4.

- **Spelling errors** are ideally errors produced from misspelling a word. In the ideal case, the language checking software also suggests the correct spelling. However, the case might also be that the word is correctly spelled but not included in the software dictionary.
- **Grammar errors** can be divided into two types (Bustamante & León, 1996), (Sågvall Hein, 1998); a) A *structural grammar errors* and *b) non-structural errors.* A *structural error* can be corrected by inserting, deleting, or moving one or

[1] http://www.asd-ste100.org/
[2] http://www.apastyle.org/

more words. A *non-structural error* can be corrected by replacing an existing word with a different one.

- **Style errors** are errors that do not fit into the spelling and grammar categories according to (Naber, 2003). Examples include catching complicated sentence structures and uncommon words. Other style related rules and errors can be associated with specific corporate language and language used in a certain genre of writing.
- **Semantic errors** (Naber, 2003) are concerned with the truth and logic of a sentence. An example of a sentence containing semantic errors is "*I love to drive my potato to the song every year.*"

In many cases, the difference between two error categories is clear. However, in some cases such as with the rule *"sentences should start with a capital letter"*, the assignment of an error category is not as clear – is it a style error, or a grammar error? In addition to the referred error types I would like to add the following error/rule types.

- **Word formation/derivation**: How should new words be constructed? This is perhaps more pertinent to languages such as Swedish where noun-verb transformation is more complex than in e.g. English. Example: *Google (N), Googla (V)*. Another example of word formation/derivation is creating nouns from proper names in Swedish: *Amerika → amerikanisera, Finland → finlandism*
- **Terminological error**: A terminological error occurs when a forbidden term is used instead of the approved term. To detect terminological errors and correct them, access to a term bank is needed. For example, when documenting a particular operating system, the term "*directory*" may be forbidden and should be replaced with "*folder*".
- **Typographic error**: Using the wrong glyph, spacing e.g. use of regular quotes rather than smart quotes, using the wrong spacing or dash glyphs, using three separate periods instead of an ellipsis glyph.

The word formation/derivation category could be grouped into the *style error* category or the *spelling* category. Whether or not it is a subcategory to or a proper category is a minor issue however. The point of including it as a separate category is because of its *productive nature*. The category deals with how new words are constructed which in essence means defining a dynamic dictionary, but also rules of style regarding concerned with why one alternative is better than another e.g. choosing "*icke-kemisk*" over "*ickekemisk*" (Eng. *Non-chemical*).

3 Information Levels

In addition to error classification I would also like to propose classification of the information level needed to detect different errors. There are two aspects of information that are relevant when implementing a language checker – *feature* and *access*. Features are attribute values that can be assigned to e.g. a token or a phrase. Access is about how many tokens can be considered by the system – a single token, a single token and its predecessor and successor, any token in a sentence, tokens from multiple sentences, tokens from the whole document?

For example, the two categories of grammatical errors previously mentioned (structural and non-structural) are good linguistic error categories, but when building a system that implements these rules, the linguistic categorization scheme is not enough. Detecting different structural grammar errors requires different features, i.e. two different kinds of structural grammar errors may need two separate feature sets.

Depending on the available *document markup*, certain features may or may not be available. In some cases, there are workarounds. For instance, even though a sentence is not marked up as being part of a numbered list, it may be possible to deduce this by looking at the first characters of the previous and following sentences. If information about whether or not a sentence is part of a list is available, the access requirement is single sentence. However, if such information is not available, the access requirement is multi-sentence. The information features and access scope levels are presented in the listing below.

3.1 Orthographic features

- characters in token: `o_token-chars`

The characters that each token is composed of is the most basic feature.

3.2 Morpho-Syntactic features

- part-of-speech of token: `m_token-pos`

- token chunks/phrases: m_chunk
- clause, sub clause: m_tree

There are of course many more linguistic features that can be considered. However, these three should suffice in most cases.

3.3 Document structure feature*s*

- lower-cased alphabetic-numbered-list: s_low-alpha-num-list
- sentence is heading: s_heading
- part of table: s_table
- list: s_list
- quote: s_quote

Document structure related features contain information on the document semantics of a token or a token sequence, e.g., if the current sentence is a heading, or a list item. Depending on where the sentence is found in the document, different rules may apply. When writing in English, capitalizing nouns is acceptable in headings but not in body text. Document structure related features are in most cases provided by the authoring environment (e.g., a word processor). Whether or not they are present in the text to be analyzed by the language checker has a huge impact on how rules need to be implemented as demonstrated in the previous example related to capitalization in headings.

3.4 Semantic features

- date: s_date
- time: s_time
- is the word a geographic proper name: s_prop-geo
- language exception, e.g., a Swedish word in English text: s_language
- word is an abbreviation: s_abbrev
- word is a contraction: s_contract
- internet link, e.g., e-mail or URL: s_internet
- lexical semantic information: s_lexsem

All semantic features except the last feature are document structure-related. The last feature, "*lexical semantic information*" is a catchall feature for information such as knowing whether the word "*bank*" refers to a financial institution, a location near a river in the sentence "*I went to the bank.*" The availability of semantic features has a huge impact on how rules can be implemented. Semantic information is usually not inferred using algorithmic analysis, but might rather be available to the language checker e.g. in a corporate CMS environment where items such as telephone numbers, part numbers etc. are explicitly marked up in the source text.

3.5 Information scope/access

- serial access to tokens, i.e. one token at the time is processed: a_serial
- random access to all information of all tokens in a sentence: a_sent-tokens
- random sentence spanning information: a_multi-sent

In the access scheme above, it is assumed that access categories include lower numbered access categories as well, e.g., a_multi-sent access includes a_sent-tokens access.

4 Brief analysis of "*Skrivregler för svenska och engelska från TNC*"

The guidelines used in this paper were taken from the Swedish handbook "*Skrivregler för svenska och engelska från TNC*"[3] ("*Rules writing Swedish and English from TNC*" (SR-TNC)). In SR-TNC there are 216 paragraphs concerning the Swedish language. However, these paragraphs may contain more than one actual rule to implement. The guidelines in SR-TNC are written with technical documentation in mind. All 216 paragraphs were considered, but not all 216 paragraphs were chosen for analysis.

Instead, a selection of 30 of the 216 paragraphs was selected. The categorization scheme was iteratively revised as the analysis progressed.

5 Examples and discussion of analyzed rules

In **Table 1** an example is given of how the categorization scheme was applied. An example of a grammar rule can be found in SR-TNC paragraph 16 which states that a comma should be used after the conjunctions *och*, *eller*, *men* and

[3] Terminologicentrum (TNC), is a Swedish organization working to improve and technical writing. TNC has published several handbooks and dictionaries and also acts as advisors in terminological issues.

src-id	rule type	feature	access	rule name	description
tnc-2	style	o_token-chars	a_sent-tokens	ingen-extra-punkt-när-mening-avslutas-med-förkortning	Om en mening slutar med en förkortning försedd med punkt utelämnas meningens avslutningspunkt.
tnc-6	typography	t_token-chars	a_sent-tokens	elips	elipstecken eller tre efterföljande punkttecken ska användas. Inte två eller fler än tre punkter vid elips
tnc-16	grammar	m_tree	a_sent-tokens	komma-efter-bindeord	komma efter och, eller, men och utan när de används för att sammanfoga två huvudsatser
tnc-40	formation	o_token-chars	a_sent-tokens	bindestreck-vid-icke-sammansättning	t.ex. icke-kemisk, icke-metaller icke-proteinkväve, ickerökare och ickevåld. Bedömningskriteriet är "tydlighet" hos ordet.

Table 1: Rule examples

utan (Eng. *and*, *or*, *but*, *without*) when they are used to connect two main clauses. This rule was categorized as a *grammar rule* information on main clauses and sub clauses in the sentence and must be able to access the tokens in the sentence in any order.

Several rules need lexical semantic information about the word, e.g., in *SR-TNC paragraph 100* which states that only when a geographical proper name refers to the actual place should it be capitalized (Swedish). For example, *Manchester* should be capitalized when it refers to the city of *Manchester*, not when it refers to the cloth *manchester*. This is hard to deduce or include in texts, but a task for Language Technology research might be to notify the writer of the possible error by e.g. asking the writer "*When you write Manchester, do you mean the city or the cloth?*".

From the brief analysis of SR-TNC, it is also clear that there is also a need for rules that describe how new words are produced. It is impossible to create a dictionary containing all possible wordforms, so perhaps a more dynamic component than the standard dictionary needs to be created. In languages such as Swedish, word compounding is one kind of productive mechanism. Efforts have been made to cope with compound words when performing spell checking (Domeij et al 1994), other kinds of out of vocabulary words must also be caught, such as inflection of foreign words and new abbreviations.

6 Conclusion

Analysis of rules for writing in handbooks is needed to develop a framework which is capable of incorporating such guidelines. The current state of language technology has not achieved a level where all rules can be implemented in a software language checker due to analysis methods, available semantic metadata and the subjective nature of certain guidelines.

However, the analysis can provide insights into which areas within language technology need further research to provide such a tool.

Regarding the implementation of rules following this categorization scheme, some trials have been done. Here a pragmatic approach was chosen adding stylistically preferred words to the dictionary. This approach is not a general approach, but even so, may be a feasible approach when considering e.g. domain specific technical documentation.

References

Almqvist, I., & Hein, A. (2000). A Language Checker of Controlled Language and its Integration in a Documentation and Translation Workflow. *Proceedings of the Twenty-second International Conference on Translating and the Computer* .

Bustamante, F. R., & León, F. S. (1996). GramCheck: A Grammar and Style Checker. *Proceedings of the 16th International Conference of Computational Linguistics (Coling –96)*, (pp. 175-181).

Domeij, R., Hollman, J., & Kann, V. (1994). Detection of spelling errors in Swedish not using a word list en claire. *Quantitative Linguistics* , *1*, 195-201.

Domeij, R., Knutsson, O., Carlberger, J., & Kann, V. (1999). Granska - an efficient hybrid system for Swedish grammar checking. *NODALIDA 1999.*

Naber, D. (2003). *A rule-based style and grammar checker.* Technische Fakultät. Universität Bielefeld.

Sågvall Hein, A. (1998). A Chart-Based Framework for Grammar Checking Initial Studies. *Proceedings of NODALIDA '98*, *11*. Center for Sprogteknologi, Denmark.