Design and evaluation of grammar checkers in multiple languages

Antje HELFRICH
NLG, Microsoft Corp.
1 Microsoft Way
Redmond, WA 98052
antjeh@microsoft.com

Bradley MUSIC
NLG, Microsoft Corp.
1 Microsoft Way
Redmond, WA 98052
brmusic@microsoft.com

Abstract

This paper describes issues involved in the development of a grammar checker in multiple languages at Microsoft Corporation. Focus is on design (selecting and prioritizing error identification rules) and evaluation (determining product quality).

Introduction

The goal of the project discussed here is to develop French, German and Spanish grammar checkers for a broad user base consisting of millions of Microsoft Word customers – users who create documents of all types, styles and content, using various terminology and dialects, and who want proofing tools that help eliminate mistakes in an efficient and non-intruding fashion.

The fact that the user base is so broad poses many challenges, among them the questions of which errors are most common among such a diverse set of users, and what types of input the grammar checkers need to be tested and evaluated on.

This paper will describe the common methods and processes that we use across the language teams for design and evaluation, while focusing on language-specific characteristics for the actual product design. The central role of large text corpora in the three languages (including regional variations) for both design and evaluation will be discussed.

Design: How do we know what the right features are?

In the design phase of the software development process, we ask the question: What should the product do for the user? For a grammar checker, the main features are the error detection and correction rules, or "critiques". The goal of the design phase is to determine which level of proofing and which error types typical Microsoft Word users care about most. It is important to remember that our grammar checker is not a standalone product, but a component within Microsoft Word, and that the main goal of the user is to create documents as efficiently as possible. We don't want to distract, delay or bother people with a picky proofing component that points out linguistic issues most users don't care about (even if we could critique those with high precision[1]) or eagerly highlights any "suspicious" sentence with a potential problem. Instead, we focus on critiques that are actually helpful to the majority of users from their point of view and support them in their ultimate goal of creating grammatically clean documents efficiently.

Researching the customer

The first step towards determining the feature set is to describe the target user of the grammar checker. One early decision was to focus on native users, since we are developing a grammar checker and not a language-learning tool. However, many of the grammar mistakes native users make are also – or even more – common among non-native users, so we know that the grammar checker will be helpful to this population as well.

The target user base for our grammar checkers are current and future Microsoft Word users, and we benefit from information that has already been gathered about the Microsoft Word user profile. We know that Microsoft Word is used mostly at the workplace, and we know what types of documents various professionals create in the respective countries.

In addition to relying on general Microsoft Word user information, we learn about people's proofing behavior in interviews, focus groups and

[1] Even actual errors can belong in this category: the French capitalization rules for language names vs. people (e.g. *français* vs. *Français*), for instance, are clear, but customer research shows that many users don't want such errors to be pointed out.

surveys, conducted in the target markets Germany (where we include Swiss and Austrian speakers), France, Canada, Spain, and Latin America. We develop discussion guides and questionnaires to gather detailed information about how people ensure that their documents are "grammar-clean", starting with questions about the types of documents they write, whether they care about the correctness of their writing equally for all documents (and have found, not surprisingly, that the level of desired proofing depends on the intended formality of the document, which in turn depends on the target audience for the text) and proceed with questions about how they proof their texts and what types of issues they feel they need help with.

Focus groups and survey participants provide a lot of input on the question of what errors people care about most. We know from these studies to focus on actual grammar errors instead of on stylistic issues, since there is no common agreement about the latter and people are generally less interested in seeing them pointed out. We also receive detailed feedback on language-specific priorities for error detection: we have learned for instance that French speakers care about agreement and getting tense and mood right, German speakers care about selection of case, capitalization and spelling together vs. apart rules, and Spanish speakers care about agreement, correct use of clitics, and confusable words, among other error types.

Selecting and prioritizing the features

After determining the target user for the grammar checker, we systematically compile the set of critiques that will be helpful to this user base. For features like the user interface we use data gained from user feedback concerning the existing English grammar checker and confirm the findings in the target countries; the actual error recognition rules, however, are selected solely on a language-specific basis.

The methods we apply in order to determine the critique sets are systematic and are shared among the teams. First, error types and potential critiques are compiled based on the sources listed below; in a second step we prioritize and trim down the list of potential critiques according to criteria of frequency, helpfulness, and reliability.

Language/linguistic knowledge: Each language team consists of linguists and computational linguists who grew up and were educated in the native language community. We painfully remember grammar rules that were drilled into us back in elementary school and have theoretical and practical experiences that range from language teaching to translation/localization backgrounds to PhDs in linguistics. While we know that disagreement errors are common in all our target languages due to forced agreement (between subject and verb or between a noun and its articles/modifiers), we pay special attention to language-specific phenomena and error types. For instance, analysis of French errors reveals a high degree of confusion between infinitive and past participial verb forms, presumably due to their phonetic equality; we therefore developed special confusable word detection algorithms for the French grammar checker.

Another aspect of language knowledge is to observe trends and changes in language use, whether the changes are speaker-induced (e.g. gradually changing case requirements after specific prepositions in German) or externally motivated like the spelling reform in Germany, which has huge consequences for the grammar checker[2].

Reference books: Books about typical (and frequent) grammar errors can be hard to come by, depending on the language being analyzed, though we did find sources for typical "grammatical stumbling blocks" for all languages. Excellent information came from books about writing good business letters, since their target readers overlap with our target users, and they contain good lists of grammar issues that people often grapple with (e.g. capitalization in multiple word expressions, including standard business letter phrases, in German). Unfortunately most of these give no (or very little) indication of the frequency of the error.

Customer research: As described above, we spend considerable time and effort to investigate what errors native language users struggle with and would like help with.

[2] The spelling reform affects the grammar checker since many changes in capitalization rules and spelling together vs. apart rules require syntactic parsing in order to identify and correct mistakes. An example is "zur Zeit" which is still spelled apart when governing a genitive object, but is, according to the new spelling rules, spelled together and with lower case ("zurzeit") when used adverbially.

Market analysis: We study the market for grammar checkers and proofing tools in general in the French/German/Spanish-speaking countries, to review what products and features users are familiar with and might expect in a grammar checker.

Text corpus: We process and review millions of sentences for each language to find out which errors actually occur and at what frequency.

All of the sources listed above contribute to the design process. The most decisive factors stem from our customer research, which informs us about what users view as their biggest grammar challenges, and the corpus analysis, which informs us about what errors users actually make. Corpus analysis plays such a central role in our feature design that it is discussed separately in the next section.

Analyzing text and error data

Our text corpora are central for product design and evaluation, and we are investing heavily in creating, acquiring, categorizing, storing, tracking, and maintaining data for the grammar checker and future product development projects. While we have to compile three separate corpora for French, German and Spanish, the methods and principles we apply to building and maintaining the corpora are shared.

The corpus used in the grammar checker project is representative of the documents that target users create, and therefore the input that the grammar checker will have to deal with. It includes a mix of documents from various media (e.g. newspaper vs. web site), styles (e.g. formal vs. casual) and content (e.g. finance vs. science). The proportion of each category is predetermined according to the Microsoft Word user profile described above.

The research community benefits from access to published corpora not available for commercial use. In contrast, a corporation that needs data for development and testing purposes is much more restricted. The following list gives an overview of some of the challenges we are faced with:

Copyright issues: While we are surrounded by a lot of text, especially on the internet, many of these documents are copyrighted and cannot be used without permission; we need to follow detailed legal guidelines and procedures, which can cause substantial lag time between identifying useful corpus sources and actually acquiring and using them.

Size: We need huge amounts of corpus, in all languages, in order to represent the various media, styles, and contents. To render test results meaningful, we need to ensure that all of the error types we develop critiques for have sufficient representation in the corpus.

Edited vs. unedited data: For our purposes, we are especially interested in text that has not undergone proofing and revision, in order to find errors people actually make while entering text, as well as to later test the quality of the grammar checker. Such documents are extremely hard to come by, so we found ways to have such unedited text data specifically created for our project. Edited data is used to verify that the grammar checker does not falsely identify errors in correct input.

Blind vs. non-blind data: We divide our corpus into two parts of equal size and corresponding content as far as document types, subject matter and writing styles are concerned. Half of the corpus is available to the whole team and is used for design and development as well as testing: The program manager uses this corpus to identify and analyze error types and frequency, and to support developers by providing corpus samples for specific grammatical constructions or error occurrences; the test team uses it to provide open feedback to developers about the precision of the parser and the grammar checker. The other half of the corpus is “blind” and only available to the test team; it is used to measure the accuracy of both parser and grammar checker. When the test team finds “bugs” (e.g. missed error identification, or faulty analysis of a correct sentence as grammatically wrong) in the blind corpus, the underlying pattern of the problem is reported, but the specific sentence is not revealed in order to prevent tuning the product to individual sentences and biasing the accuracy numbers. Doubling the corpus in this way means that we need more data in terms of sheer quantity; it also poses additional challenges for categorizing, tracking, and securing the data.

Cleaning: While we don’t want the corpus clean in terms of grammar errors, we do need to process it to standardize the format, remove elements like HTML formatting codes, hard returns, etc. so we can use it in automated tools.

The extensive effort put into design helps to ensure that the product focuses on errors people actually make and care about. The next section describes the testing done to determine if we've achieved acceptable quality for identification and correction of these error types.

Evaluation: How do we know when we're done?

During the development process, testers give feedback and quality assessment, based on both the blind and non-blind corpora, and using a variety of tools to provide quick turn-around after a change to the system. Development feedback shows the effects of each change to the lexicon, morphology, grammar or critiquing system, where the testers systematically apply language-independent methods of analysis and reporting. Developers need to know the impact of any changes they make as soon as possible, so that further development can proceed with confidence or, in the case of an unexpected negative impact, problems can be corrected before further development. Quality assessment is partially reflected in terms of agreed-upon metrics, such as recall, precision, and false flags per page.

As we approach the end of the development process, we continue to monitor the metrics against pre-defined goals, but also shift focus to other kinds of testing with orientation towards the user's experience with the grammar checker.

This section will briefly outline key metrics used for quality assessment as well as some of the user-focused testing we do before shipping the final version.

Precision

Precision = good flags/total flags, e.g. if the grammar checker correctly identifies 160 errors on a given corpus, and incorrectly flags 15 words/phrases as errors in that corpus, the precision will be 160/175=91%. Determining precision has less meaning the more the test corpus has undergone editing. In the extreme case of a highly edited text (e.g. a published book) where in principle there should be no grammar errors present at all, the only flags a grammar checker could possibly get would be false flags, thus precision would be 0%, which would give an inaccurate impression of product quality.[3] Precision is reported on a variety of corpora within the language teams, these having the same representativity across the teams.

Recall

Recall = good flags/expected flags, i.e. what percentage of the errors is actually spotted. Research on users' impressions of grammar checker quality consistently shows that users are less concerned about recall than about the number of false flags. This has entailed a cross-linguistic prioritization of improving quality by the reduction of false flags. In terms of metrics, this means that increasing precision and decreasing the false flag per page rate have had a higher priority than recall for these grammar checkers. One challenge here is the fact that methods for reducing false flags can risk loss of good flags that would be helpful to our users, so a light hand is required to balance reduction of the absolute number of false flags vs. still spotting and correcting the errors people really make.

False flags per page

Although highly edited texts are less interesting for determining precision, they are important as a basis for measuring how 'noisy' the grammar checker is on a finished document.[4] This can be measured in terms of false flags per page, with the ideal being zero – however language being as complex as it is, it is in fact extremely difficult to achieve no false flags in a system that attempts to parse and correct the frequent errors in agreement, mood, etc. More realistically, a trade-off has to be accepted that gives the critiques room to work, while still staying under what's considered an annoying level of false flags per page. In the French, German and Spanish grammar checker development effort, we set a goal

[3] This was a flaw in a recent evaluation of a French grammar checker done by the French Academy, where a grammar-checking product was run against French literature from the last four centuries, with the none too surprising result that it suggested changes to the great authors' prose. [AFP99]

[4] Note that noisiness is affected by factors other than grammar checker quality; for instance the UI can help to reduce annoying flags by remembering editing of each sentence so as not to bother users with same errors once they've been explicitly ignored, as is done in Microsoft Word.

of having less than one false flag per page. Once we were well below that for each language (while still achieving precision and coverage goals), we subjected the grammar checkers to beta testing (see below) to confirm whether the users' impressions of the helpfulness of the grammar checker conform to the metrics.

Market analysis

Although the Natural Language Group doesn't sell the grammar checkers as standalone products, it is still interesting for us to determine how we fare against grammar checkers already on the market. Since we can't be sure that other grammar checkers have been evaluated in exactly the same way, we can't rely on the competitors' reported metrics, such as false flag per page rates. We therefore do our own objective quality comparison based on the same *blind* corpus as we evaluate ourselves against. Here is where the strict division between blind and non-blind is absolutely essential to avoid skewing the results – if the non-blind corpus were used, we would show ourselves to an advantage, since the developers also have access to that corpus and naturally train against it.

Final testing: 'real world', bug bashes, beta testing

Even given a low false flag per page rate and acceptable precision and recall measures, when all is said and done the user's impression of quality and usefulness can still come down to highly specific contexts. Regardless of how we score on our own metrics, users will often turn a grammar checker off due to a 'spectacular' false flag and/or annoyance. An example of what is meant by a spectacular false flag is *Nous sommes* ➔ **Nous somme*, where *sommes* is the correct first person plural present form of the French verb *être* 'to be', while *somme* without the *–s* has both masculine and feminine noun readings, entailing the possibility of a misparse of the verb as a noun, and therefore a potential false flag. A Canadian user who encountered this false flag when using a product that is now off the market immediately turned off that grammar checker for good. An error on a common word like this can give users a very low opinion of the grammar checker quality and cause them turn it off for this flag alone. Regardless of what the metrics tell us as to overall quality, it still comes down to a subjective user experience.

Therefore, when the product is getting close to its final shippable state, we break away from the metrics to gain insight into how users experience the grammar checker quality and usefulness. 'Real world testing' refers to test passes where the test teams use the grammar checker to edit documents like actual users will. Rather than focusing on detailed analysis of specific errors, they gain a general impression of the product quality.

Bug bashes are another type of testing, where native speaker users from outside our group are asked to set aside a dedicated time to finding bugs in the grammar checker. These normally take place over several hours. Users may be asked to explore the limits of the grammar checker, for instance by executing certain tasks, such as proofing an existing document, changing their settings, etc. The purpose is to find functional and linguistic bugs that may have been missed by our own extensive testing. We also ask the participants to answer a few questions on their overall experience with the grammar checker.

Finally, beta testing is simply where the grammar checker is used in native speakers' daily document production environment – they are asked to use it in their daily work, submitting bugs via email. Eventually they are also asked to record their impressions on the usefulness of the grammar checker, with as many specifics as possible.

Conclusion

Developing a grammar checker for a broad user base presents many challenges, and this paper focused on two areas: design and evaluation. The multilingual project environment allows for substantial leveraging of knowledge, methods and processes; in the end, though, a grammar checker's value is determined by the support it provides to a specific language community. To this end, native language data guides the development, including the analysis of large corpora and intense study of the target market's customer and their proofing needs.

References

[AFP99] Agence France-Presse press release, May 21, 1999, "L'Académie française met en garde contre des logiciels de correction".