Suggestion prioritization mechanism

The suggestion prioritization mechanism is designed to allow the admins to influence and change the default behavior of how the corrections are generated and shown to end users. Using this mechanism, the admins can create a list of rules which will be used during the process of generating corrections.

Here is a list of use cases where the suggestion prioritization mechanism will come in handy:

- Common spelling errors (e.g. “teh” –> “the”)
- Synonyms and alternatives

Rules categories

There is a list of main categories that can be used to create your custom rules for the suggestion prioritization.

Category name	Description
ANY_POSITION	Any position in a word
WORD_INITIAL	The first letter in a word
FIRST_OR_SECOND	The first or the second letter in a word
MIDDLE_OF_THE_WORD	Not the first and not the last letter in a word
END_OF_THE_WORD	The last letter in a word (end of a word)
ENTIRE_WORD	The entire word

Syntax and configuration File

All the rules should be written in a special *.ini configuration file.

Configuration file

The configuration file for the suggestion prioritization mechanism must match the next requirements.

- File format: *.ini
- Encoding: UTF-8
- Target location: [WebSpellChecker_Installation_Folder]/AppServer/
- Example of file name: “suggestions_prioritization_rules_lang_shortcode.ini”. Here is a list of all available languages with their short codes.

You can have more than one configuration file with rules for different languages. Also, you can reuse a single file with rules for a list of languages which might have the same rules (e.g. English dialects).

Syntax

The syntax of the configuration file with rules for the suggestion prioritization must follow the next requirements.

- Rules blocks must start with the category name.
- Specific rule follows after the category name to which it belongs. You can have more than one rule under a single category.
- Each rule should be written on a new line.
- The rule consists of two values separated with an empty space: [value 1] [space] [value 2]

```
CATEGORY_NAME
[value 1] [space] [value 2]
[value 1] [space] [value 2]
...
CATEGORY_NAME
[value 1] [space] [value 2]
...
blank line
```

⚠ If you are using the ENTIRE_WORD rule, make sure that all words that you use are present in the language or custom dictionaries.

This is an example of configuration file syntax with two rules:

- End of the word. We added two letters separated with a space: “z” and “s”. This rule will word for all misspelled words which end on “z”, and provide similar suggestions which end on “s”.
- Entire word. In this rule we replace the entire word with the other separated with a space: “teh” and “the”. Thus, when user types “teh”, the system will suggest correcting it to “the”.

```
END_OF_THE_WORD
z s
ENTIRE_WORD
teh the
```

Getting started

Once a new configuration file with the rules is ready and placed in the [WebSpellChecker_Installation_Folder]/AppServer/ folder. The next step is to enable the suggestion prioritization mechanism in the AppServerX.xml file.

1. Stop AppServer.

2. Go to the [WebSpellChecker_Installation_Folder]/AppServer/ folder and open the **AppServerX.xml** file.

3. In the list of all available parameters for AppServer, locate the **SuggestionsPrioritization** (previously CustomSuggestionsPrioritization) tag which is responsible for the suggestion prioritization mechanism. By default, the tag is commented.

4. Uncomment the section with the CustomSuggestionsPrioritization tag and fill in the information in the sub-tags as follows:

- **LangID (previously Lang**) – this parameter sets the short code of the language for which the suggestion prioritization mechanism with the defined configuration file will be applied.
- **ConfigFile (previously ConfigFileName**)– this parameter specifies the name of the configuration file which will be used for the suggestion prioritization mechanism.
- **AdvancedSearch** – this parameter enables the advanced search of the suggestions using not only the initial suggestion list given by the engine but also the language dictionaries, global custom and personal user dictionaries.

```
...
<SuggestionsPrioritization>
     <PrioritizationEnable>true</PrioritizationEnable>
<!-- Rules for each language -->
     <PrioritizationLang>
     <LangId>en_US</LangId>
     <ConfigFile>suggestions_prioritization_rules_en_US.ini</ConfigFile>
     <AdvancedSearch>true</AdvancedSearch>
     </PrioritizationLang>
</SuggestionsPrioritization>
...
```

If you want to use the suggestion prioritization mechanism for several languages and have separate configuration files, you need to duplicate the PrioritizationLang tag (see example below).

```
...
<SuggestionsPrioritization>
     <PrioritizationEnable>true</PrioritizationEnable>
      <!-- Rules for each language -->
     </PrioritizationLang>
        <LangId>en_US</LangId>
        <ConfigFile>suggestions_prioritization_rules_en_US.ini</ConfigFile>
        <AdvancedSearch>true</AdvancedSearch>
     </PrioritizationLang>

     <PrioritizationLang>
         <LangId>ar_SA</LangId>
         <ConfigFile>suggestions_prioritization_rules_ar_SA.ini</ConfigFile>
         <AdvancedSearch>true</AdvancedSearch>
     </PrioritizationLang>
</SuggestionsPrioritization>
...
```

5. Once all the changes are done, save the file and start AppServer.