

Enabling AI engine as grammar provider

With release v5.7.0, we introduced an enhanced AI-based proofreading engine for English. It was built using modern approaches in Natural Language Processing (NLP). Unlike the classical rules or dictionary-based engines, it's based on the huge pre-trained language model. Its accuracy is at least three times better.

However, at the same time, there are some **cons** that should be taken into consideration:

- It is available for general English. It doesn't distinguish dialects for now. Thus, offered as a separate language.
- Suggestions can be somewhat unpredictable and inconsistent. Output of the neural network depends a lot on the context and is very sensitive to even slight changes in the sentence.
- There is no option to ignore a rule as all suggestions are generated with unique IDs.
- AI technology consumes more resources to provide the results. Its response time is longer compared to classical engines. Check the updated [Hardware requirements for v5.7.0+](#).
- It is available only for modern products, such as the WProofreader add-on for rich text editors and API.
- There are no types or differentiation by error type (spelling, grammar, style, punctuation). All are combined in one. Later we will add types.

You can give it a try using one of the public demos on our website: <https://webspellchecker.com/wsc-proofreader/>

If you are using the server-based package of WebSpellChecker v5.7.0 and higher, then you have an AI engine included in the package.

AppServerX.xml

```
<Language Id="en_AI">
  <Alias>ai</Alias>
  <GrammarCheckProvider>neural</GrammarCheckProvider>
  <GrammarCheckProviderOptions>en</GrammarCheckProviderOptions>
  <!--<ThesaurusEnabled>true</ThesaurusEnabled>
  <SpellEngineOptions>
    <Locale>am</Locale>
    <SpellCheckProvider>ssce</SpellCheckProvider>
    <Dictionary FullPath="ssceam2.clx" >
      <ForSuggest>no</ForSuggest>
    </Dictionary>
    <Dictionary FullPath="ssceam2s.clx">
      <ForSuggest>yes</ForSuggest>
    </Dictionary>
    <Dictionary FullPath="keywords.clx"></Dictionary>
    <Dictionary FullPath="ssceam.tlx"></Dictionary>
  </SpellEngineOptions-->
</Language>
```

AI-based engine as grammar provider for American English

You can enable an AI-based engine and use it as a grammar provider for one of the existing dialects, e.g. American English.

1. Stop AppServer before making any changes in AppServerX.xml.
2. Go to AppServerX.xml file and open it for editing.
3. Scroll down to the section with language tags. Locate the language tag <Language Id="en_US">...</Language> for American English.



The same approach can be used for any other available dialects of English, e.g. British or Canadian English.

AppServerX.xml

```
<Language Id="en_US">
  <Alias>en</Alias>
  <Alias>am</Alias>
  <GrammarCheckProviderOptions>en-US</GrammarCheckProviderOptions>
  <ThesaurusEnabled>true</ThesaurusEnabled>
  <SpellEngineOptions>
    <Locale>am</Locale>
    <SpellCheckProvider>ssce</SpellCheckProvider>
    <Dictionary FullPath="ssceam2.clx">
      <ForSuggest>no</ForSuggest>
    </Dictionary>
    <Dictionary FullPath="ssceam2s.clx">
      <ForSuggest>yes</ForSuggest>
    </Dictionary>
    <Dictionary FullPath="sscema2.clx"/>
    <Dictionary FullPath="keywords.clx"/>
    <Dictionary FullPath="ssceam.tlx"/>
  </SpellEngineOptions>
</Language>
```

4. Add an extra tag, **<GrammarCheckProvider>neural</GrammarCheckProvider>**, inside the language tag. The value should be **neural**.

AppServerX.xml

```
<Language Id="en_US">
  ...
  <GrammarCheckProviderOptions>en-US</GrammarCheckProviderOptions>
  <-- Enabling AI-based engine as a grammar check provider. -->
  <GrammarCheckProvider>neural</GrammarCheckProvider>
  ...
</Language>
```

5. Fix the default value inside **<GrammarCheckProviderOptions>...</GrammarCheckProviderOptions>** from **en-US** to **en**.

AppServerX.xml

```
<Language Id="en_US">
  ...
  <GrammarCheckProviderOptions>en</GrammarCheckProviderOptions>
  <-- Enabling AI-based engine as a grammar check provider. -->
  <GrammarCheckProvider>neural</GrammarCheckProvider>
  ...
</Language>
```

6. Start AppServer to apply changes.

Mixed grammar check provider: classical grammar check and AI-based engines

There is an option that also allows using two grammar checking engines at the same time. Though one engine suggestion will take precedence over the other.

Again, all the changes should be made in the AppServerX.xml file.

1. Stop AppServer before making any changes in AppServerX.xml.
2. Go to AppServerX.xml file and open it for editing.
3. Locate **<GrammarCheckEngine>...</GrammarCheckEngine>** tag. Inside this tag you will find **<GrammarCheckProvider>...</GrammarCheckProvider>** tag. Set the default value to **mixed**.

```

<GrammarCheckEngine>
  <!-- The grammar provider, can be one of langtool (default), atd, neural, mixed. -->
  <GrammarCheckProvider>mixed</GrammarCheckProvider>
  ...
</GrammarCheckEngine>

```

4. Uncomment **<MixedProvider>...</MixedProvider>** and inside **<Providers>...</Providers>**, specify **neural,langtool** separated by comma.

```

<GrammarCheckEngine>
  <!-- The grammar provider, can be one of langtool (default), atd, neural, mixed. -->
  <GrammarCheckProvider>mixed</GrammarCheckProvider>

  <MixedProvider>
    <Providers>neural,langtool</Providers>
    <!-- <CustomPrioritiesFile>CustomPriorities.xml</CustomPrioritiesFile> -->
  </MixedProvider>
  ...
</GrammarCheckEngine>

```

5. Now you can apply the mixed approach with the grammar engines to one of the languages.

Let's try to enable the **mixed** engines for American English. for Scroll down to the section with language tags. Locate the language tag **<Language Id="en_US">...</Language>** for American English.

6. Add an extra tag, **<GrammarCheckProvider>neural</GrammarCheckProvider>**, inside the language tag. The value should be **mixed**.

AppServerX.xml

```

<Language Id="en_US">
  ...
  <GrammarCheckProviderOptions>en-US</GrammarCheckProviderOptions>
  <-- Enabling a few engines as grammar check providers. -->
  <GrammarCheckProvider>mixed</GrammarCheckProvider>
  ...
</Language>

```

7. Fix the default value inside **<GrammarCheckProviderOptions>...</GrammarCheckProviderOptions>** from **en-US** to **en**.

AppServerX.xml

```

<Language Id="en_US">
  ...
  <GrammarCheckProviderOptions>en</GrammarCheckProviderOptions>
  <-- Enabling a few engines as grammar check providers. -->
  <GrammarCheckProvider>mixed</GrammarCheckProvider>
  ...
</Language>

```

8. Save changes in AppServerX.xml file and **start AppServer**.

Now you will have the results generated by two engines as green underlines.