Subject: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)
Posted by slashupp on Tue, 27 Nov 2018 12:37:49 GMT
View Forum Message <> Reply to Message

(linux, latest svn Upp)

How do I enable/set up spell-checking with RichEdit?
Am specifically using a UWord-derived control and want to have it automatically do spell-checking,
the spellcheck bool = true, so I guess it is a dictionary issue? how do I specify that for english US or UK?
What must I do?

(just now saw I asked this before & forgot)
anyway, what I did was:
downloaded the .udc from http://sourceforge.net/projects/upp/files/SpellerDictionaries/. as per dolik
actually from: https:// sourceforge.net/projects/upp/files/SpellerDictionaries/Aspel l/
then I clicked the language-tool in UWord toolbar & specified EN GB in the little popup-dialog

Now I get red-underlined miss-spelled words. GOOD, NICE.

Follow-on question:
When I right-click I want to display for select&replace the possible correct spellings
how do I do that?

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)
Posted by mirek on Thu, 03 Jan 2019 08:47:55 GMT
View Forum Message <> Reply to Message

Quote:
When I right-click I want to display for select&replace the possible correct spellings
how do I do that?

Thats not implemented (yet?). But it would be fun to add... Any pointers on suitable algorithm?

Mirek

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)

Hello Mirek

To measure the "difference" between words to see what are the best matches in dictionary, I use the Levenshtein distance and the DamerauLevenshtein distance. One is faster and the other, better.

As the implementation considers all characters as char, prior to this I normalize accented and special characters.

This is the implementation:int LevenshteinDistance(const char *s, const char *t) {
```
 int lens = int(strlen(s));
 int lent = int(strlen(t));

    Buffer<int> v0(lent + 1);
    Buffer<int> v1(lent + 1);

    for (int i = 0; i <= lent; ++i)
        v0[i] = i;

    for (int i = 0; i < lens; ++i) {
        v1[0] = i + 1;

        for (int j = 0; j < lent; ++j) {
            int deletionCost = v0[j + 1] + 1;
            int insertionCost = v1[j] + 1;
            int substitutionCost;
            if (s[i] == t[j])
                substitutionCost = v0[j];
            else
                substitutionCost = v0[j] + 1;

            v1[j + 1] = min(deletionCost, insertionCost, substitutionCost);
        }
        Swap(v0, v1);
    }
    return v0[lent];
}

int DamerauLevenshteinDistance(const char *s, const char *t, int alphabetLength) {
 int lens = int(strlen(s));
 int lent = int(strlen(t));
 int lent2 = lent + 2;
 Buffer<int> H((lens+2)*lent2);

    int infinity = lens + lent;
    H[0] = infinity;
```

```
    for(int i = 0; i <= lens; i++) {
  H[lent2*(i+1)+1] = i;
  H[lent2*(i+1)+0] = infinity;
    }
    for(int j = 0; j <= lent; j++) {
  H[lent2*1+(j+1)] = j;
  H[lent2*0+(j+1)] = infinity;
    }
    Buffer<int> DA(alphabetLength, 0);

    for(int i = 1; i <= lens; i++) {
      int DB = 0;
      for(int j = 1; j <= lent; j++) {
        int i1 = DA[t[j-1]];
        int j1 = DB;
        int cost = (s[i-1] == t[j-1]) ? 0 : 1;
        if(cost == 0)
         DB = j;
        H[lent2*(i+1)+j+1] =
          min(H[lent2*i     + j] + cost,
              H[lent2*(i+1) + j] + 1,
              H[lent2*i     + j+1] + 1,
              H[lent2*i1    + j1] + (i-i1-1) + 1 + (j-j1-1));
    }
      DA[s[i-1]] = i;
    }
    return H[lent2*(lens+1)+lent+1];
}I hope this helps.
```

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)
Posted by mirek on Sat, 19 Jan 2019 15:48:30 GMT
View Forum Message <> Reply to Message

First iteration commited. Be aware that you need 'new' speller files with .udc extension for this to work, you can download those here:

https://sourceforge.net/projects/upp/files/SpellerDictionari es/Aspell/

BTW, LevenshteinDistance is both too slow and does not suggest good words (e.g. for "tomrw" I expect to have "tomorrow" as the best choice and LevenshteinDistance does not lead to this), so I had to invent my own algo there...

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)

Posted by koldo on Sat, 19 Jan 2019 16:46:13 GMT
View Forum Message <> Reply to Message

Excellent!

Is the new API accessible out of RichEdit?

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)
Posted by mirek on Sat, 19 Jan 2019 18:51:24 GMT
View Forum Message <> Reply to Message

koldo wrote on Sat, 19 January 2019 17:46Excellent!

Is the new API accessible out of RichEdit?

Vector<String> SpellerFindCloseWords(int lang, const String& w, int n);

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)
Posted by koldo on Sun, 20 Jan 2019 13:47:32 GMT
View Forum Message <> Reply to Message

OK. And to get the distance between words it would be WordDistanceTester::Get().

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)
Posted by mirek on Sun, 20 Jan 2019 15:14:39 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 20 January 2019 14:47OK. And to get the distance between words it would be WordDistanceTester::Get().

Yes. It is class as some things are reused between tests... (to be more specific, I do not want clear 65536 bytes to zero after each test...).

Mirek

Subject: Re: How to add/enable spell-CORRECTING with richedit (not-quite-solved-sort-of)

Posted by koldo on Sun, 20 Jan 2019 15:20:19 GMT
View Forum Message <> Reply to Message