The SMS discrimination

Do you speak a cheap language or an expensive language ?

Notes:
In the following I will mostly refer to texts written in Latin and Cyrillic scripts, as used in Europe; however, the problem described here may apply to other scripts as well, all over the world.
In this document, by "writing correctly" a language, I mean using all national characters of that language, including accented characters, where applicable.

In today alphabetic writing systems,
each of the A, A [1], Å or Ă letters is just another letter.
None is more or less special than the other.

Sending a SMS text message over a GSM network has become a trivial practice for most of us. Although relatively cheap, sending a SMS text message still has a cost. The problem is that this cost is different, depending on the language used by the sender – even by the same sender, using the same device.

If your native language happens to be one from the "western" part of Europe, good for you: you can write correctly your language, without spelling or grammar restrictions. However, if your native language happens to be one form the "eastern" part of Europe, bad luck: if you want to write correctly your language, it costs you more.

The difference comes from the way the SMS part of the GSM standard has been originally developed: the GSM character set only covers a few so-called Western languages. Any message written correctly in any non-Western language may double or more the cost of that message. The only way of keeping the SMS text message cost at "normal" price in any situation when using the Latin script is to use exclusively non-accented characters, thus dropping any language-specific character from the text. This may lead to language crippling, for the solely reason of improper SMS technical protocol implementation and/or improper operator charging mechanism.

A Short Message Service (SMS) text message is sent over a GSM network as a stream of 1120 bits of data. The 3GPP TS 23.038 technical specification, which defines the language-specific requirements for GSM, describes three methods to represent an alphabet:

- default alphabet, 7 bits per character; a nice presentation summary of this character set can be found at the Connection Software web site [2]; it includes:
  - the basic character set (pure ASCII)
  - a few accented characters suitable for a limited set of Western languages (like French or German)
  - a few characters of the Greek character set (capital letters only)
- 8 bit data, user defined
- UCS-2 encoding, 16 bits per character

When sending a message that includes any character (accented or not) that happens to fall into the default alphabet, everything is "normal".
In this mode, a single SMS text message can include 160 characters (1120 divided by 7).

The problem appears when sending a message that includes one or more regional language characters that are outside the default alphabet. The presence of one or more of these characters will trigger the whole message in UCS-2 fixed encoding, i.e. two bytes for each single character from the Unicode base plane.
In this mode, a single SMS text message can include only 70 characters (1120 divided by 16).

(Annex A at the end of this document shows an example of how a SMS related PC application treats different characters of different languages)

This may lead to some bizarre count and pricing for messages containing same number of characters, but different national character types. Here is an example:

a. if I send the following message in Romanian language written incorrectly (with no diacritical marks at all)
   Cristian Secara butoneaza calculatorul in loc sa iasa afara la plimbare
   it will cost me the price of 1 SMS text message
b. if I send the following message in Romanian language written correctly (with all the diacritical marks on their places)
   Cristian Secară butonează calculatorul în loc să iasă afară la plimbare
   it will cost me the price of 2 SMS text messages
c. if I send the following message in French language written correctly (with all the diacritical marks on their places)
   Après-midi le soleil s'impose très largement au sud où les températures grimpent
   it will cost me the price of 1 SMS text message

As can be seen, attempting to respect the culture for one language increases the number of the messages required and the final price, while attempting the same for other language does not. Actual messages number and resulted cost may vary, depending on the particular language, number of characters, characters used, etc..

I agree the fact that in practice many users will never write accented characters in SMS text messages using their phone device keyboard (especially on those phones with “traditional” keypad), but this discussion here is for those who are willing to do that and are discouraged because of the unjustified hassle. More than that, some may wish to send SMS text messages from their computer using the PC application that came with the mobile device, in which case writing correctly a native language may be simply natural.

A proper solution to this problem is a complex matter, but first of all there should be desire to start to find one. At first glance it seemed that some solutions may exists, but none truly satisfy the discrimination problem raised here, except probably the last from the ones listed below:

- the network operators should charge based on the actual number of human-readable characters, not the number of machine low-level bytes;
  problems:
  - usually the network operator just carry an abstract data stream and charge a certain amount for the stream as a whole, without looking at what that stream contains
  - in some cases the constraint has nothing to do with the network operator, being just local and physical, without necessarily implying extra messages and hence additional costs; one example I know is the Samsung PC Studio application, where I was never able to send more than one single limited message at once, because the application does not seem to be able to generate concatenated multipart messages
- *or,*
  the mobile devices should implement the “Compression algorithm for text messaging services” as described in the 3GPP TS 23.042 technical specification;
  problems:
  - not a true i18n approach
  - the compression is limited to a handful of common words
  - the entered words must match exactly the ones from dictionary, so a keyboard typo can ruin the dictionary-based compression

    - the writing language must be known at the time the message is sent, which may or may not be the case (the writing language is not necessary related to a keyboard layout, or a given keyboard layout may cover several languages at once; the writing language is not necessary known by the system when writing the message on PC; there is no way in the world the PC will know what language I have in my mind when entering characters by using the dead keys mechanism on my PC keyboard; usually the language cannot be predicted when copy & paste some text from elsewhere)
- *or,*
  make use of the national language shift tables concept that was added since the 3GPP TS 23.038 release 8 technical specification;
  problems:
    - not a true i18n approach
    - it brings the world back to the pre-Unicode dark ages when almost each language needed its own encoding which were almost impossible to predict in every detail and when proper text exchange using national characters between users located in different part of the world proved to be doomed to failure
    - the standard requires explicit request from each country for the national table to be included, in other words writing correctly in native language needs a special request, instead of offering by default the possibility for all users of the world to write natural and correct their language if so desired
    - the writing language must be known at the time the message is sent, which may be or not the case (as stated earlier, the writing language is not necessary related to a keyboard layout, or a given keyboard layout may cover several languages at once; the writing language is not necessary known by the system when writing the message on PC; there is no way in the world the PC will know what language I have in my mind when entering characters by using the dead keys mechanism on my PC keyboard; usually the language cannot be predicted when copy & paste some text from elsewhere)
    - because of the escape characters used to trigger the switching of the national language table, the output will never reach the 160 characters per base message from which benefit the “western” laguages, so the discrimination still persist
- *or,*
  the mobile devices should be able to provide a compression scheme that handles true Unicode character encoding, like the SCSU [3]
  problems:
    - SCSU being an 8 bit encoding, the output will never reach the 160 characters per base message from which benefit the “western” languages, so the discrimination still persist
- *or,*
  upgrade the low level technology responsible with the SMS text messages with a better one, still GSM-based, that is able to treat equally all languages of the world; considering the most recent developments in communications and mobile devices industry, the actual form of SMS is simply ridiculous, from every perspective

The fundamental idea is that all languages of the world must be treated equally by any technology. It is simply absurd to consider an alphabet that begins with ABC to be more important that an alphabet that begins with АБВ, or to consider characters Å or Ñ to be more important than characters Ă or Č.

In the summer of 2008 I complained about this issue to the Commissioner for Multilingualism in the European Commission. After a while I received the answer which I quote below in italics.

> Note: at that time my report also included an issue strictly related to the Romanian language (the ş and ţ issue described on my web site [4]; I will take the liberty to omit the answer related to that issue, as it is of no relevance for the scope of this article.

Dear Mr Secară,

I would like to thank you for drawing the attention of the Commission to annoyances you are confronted with the use of language specific accentuated characters on the GSM network. As you clearly describe it, the problem only occurs in specific cases, however the promotion of multilingualism is at the heart of the priorities of Commissioner Orban.

You have correctly identified the dual nature of the problem: one being technical with the inadequate handling of 2 specific characters for the Romanian language, the other being the charging mechanisms for SMS.

[...]

For the latter issue, charging plans are entirely under the responsibility of telecom operators. The charging plans for SMS are subject to competition pressures between operators, the result being that such costs are driven down. Commissioner Reding is also applying continuous pressure on mobile operators to reduce costs of SMS. Without entirely removing the annoyances you mentioned this would at least minimize their impact. On the other hand in our Annual Information Society Report 2008 we identified that information society developments in Romania were still at an early stage with the resulting benchmarking indicators being close to the bottom of the EU rankings. We are continuing our efforts to encourage Member States to reduce the gaps between high and low performers.

I would like to once again thank you for your detailed and accurate report.

With my best wishes for your efforts

Anne Bucher
Head of Unit INFSO-C1
"Lisbon Strategy and i2010"
Directorate General Information Society and Media
Office: BU25 01/131
European Commission

I thank Anne Bucher very much for the kind response, but I consider it not good enough.

It seems I was not very clear in expressing my idea to the European Comission, but I consider the SMS text message issue to be under the responsibility of the mobile device manufacturers, not the mobile operators. It is the mobile device and the PC software accompanying the mobile device that makes the difference between, for example, Latin alphabet as used by (say) French language and Cyrillic alphabet as used by (say) Bulgarian language. It is the device that changes accordingly the maximum number of characters per SMS text message in that session. It is the device that truly makes the discrimination and in the end has a cultural and economic impact over the user.

I suppose that an electronic device must meet some criteria in order to be allowed for commercial distribution in UE market; apart from not being toxic, not inflamable, etc., it should also not be culture discriminant. Why not ?

At least in Europe, I expect that trends of discrimination caused by multilingualism to be regulated also by the European Commission. I find hard to believe (although not impossible) that all GSM operators from non-Western language countries will rush to change their charging mechanism just for cultural reasons.

On the other hand I also believe that the issue described here is rather caused by the lack of interest in internationalization matters from those who originally set the GSM technical specifications (and also the fault of those who later did too little to correct properly the technical problems or weaknesses).

Users should be able to choose freely how to treat their language, either mistreated or respectful, but either case should not be constrained by poorly designed technology. Nowadays the technical things are enough advanced to be able to cope successfully with any cultural demand.

Cristian Secară

First publish on 1 February 2011 [5], underwent some subsequent revisions

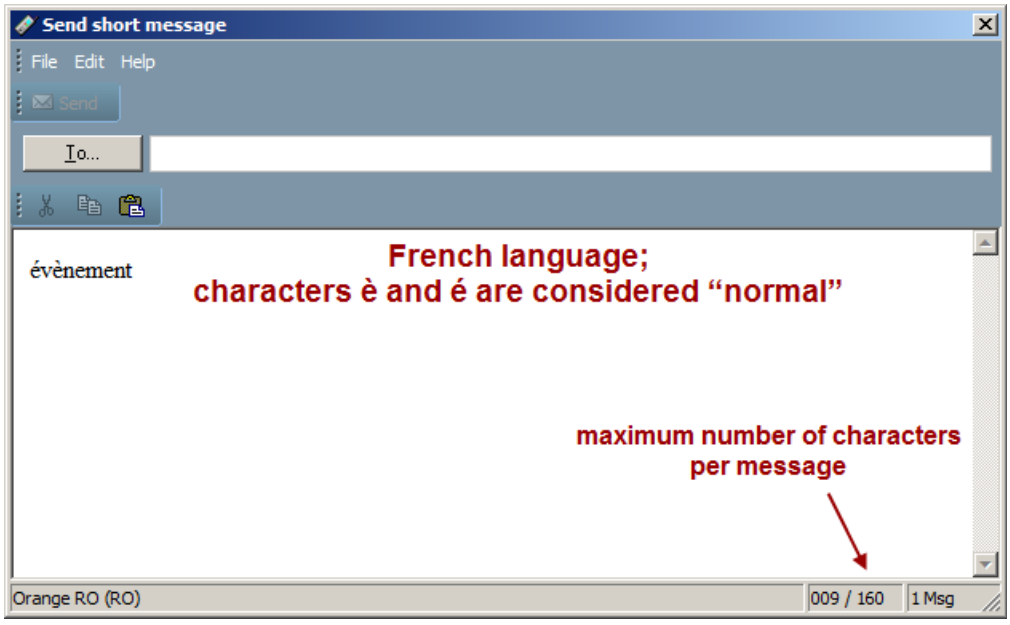
---

[1] Unicode 0x0410, i.e. the first letter of the Cyrillic alphabet
[2] http://www.csoft.co.uk/sms/character_sets/gsm.htm
[3] http://en.wikipedia.org/wiki/Standard_Compression_Scheme_for_Unicode
[4] http://www.secarica.ro/index.php/rou/s-uri-si-t-uri
[5] http://www.secarica.ro/index.php/eue/sms-story/the-sms-discrimination

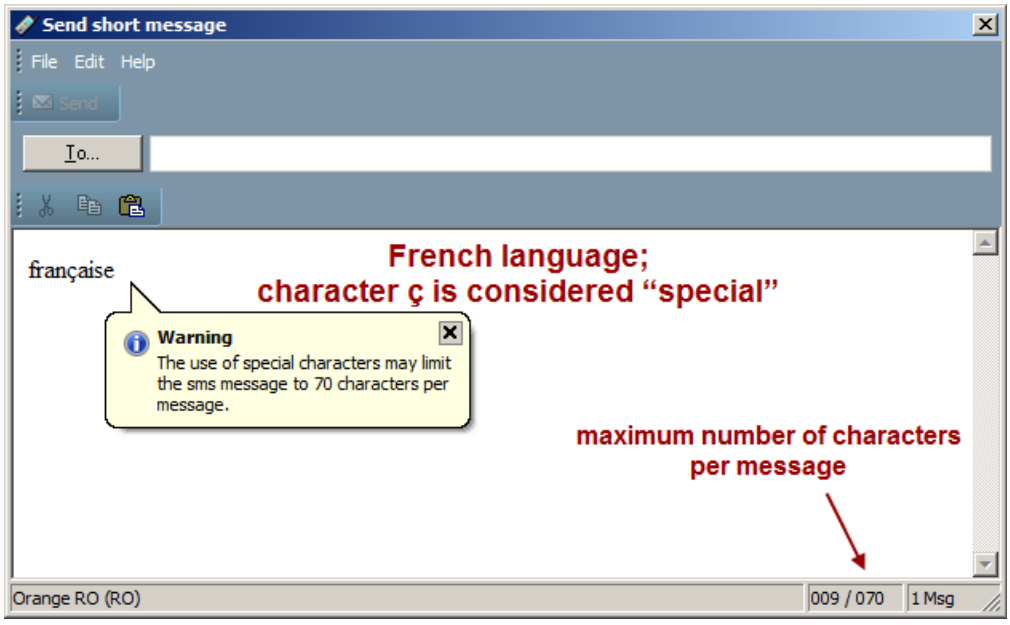
Annex A

Example of how a SMS related PC application treats different characters of different languages:

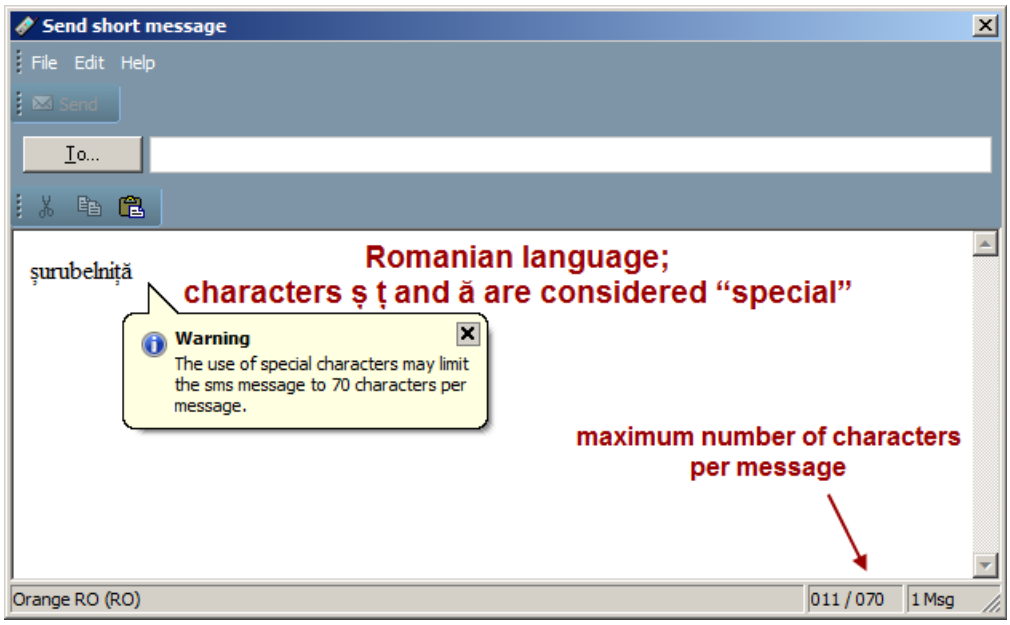
Notes:
All snapshots were taken with the locale of the PC set to my language, Romanian.
I used Motorola Phone Tools application for this exemplification only because it has a very intuitive interface in respect to the subject discussed here.



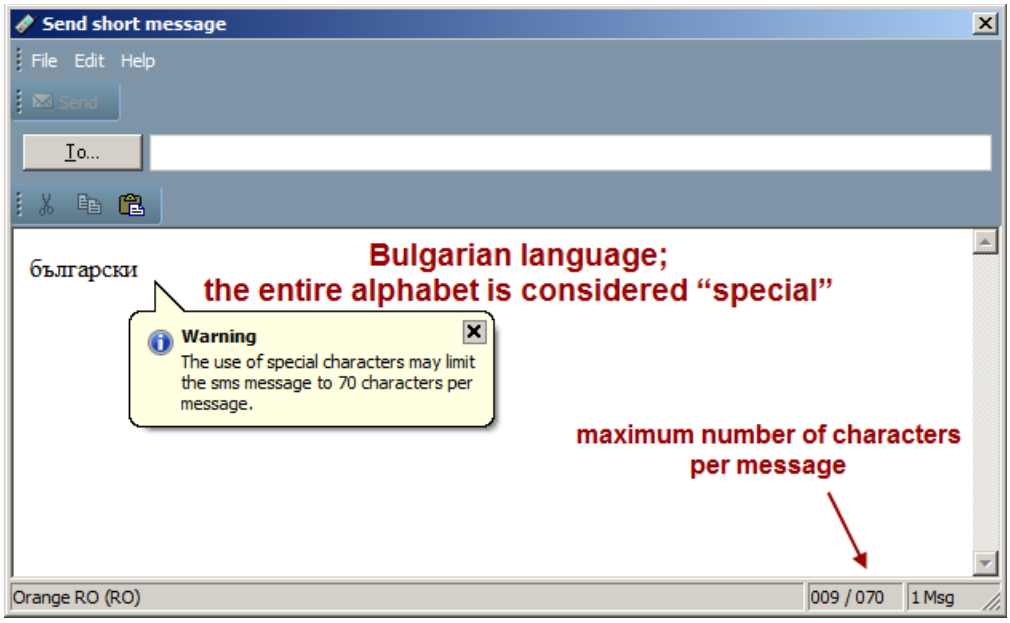
Message composition example with Motorola Phone Tools. Nothing is unusual here.



Same message, same language. What is so "special" here by comparing with the previous ?



Motorola Phone Tools thinks that ș, ț and ă are "special". According to whom ?



Now every character is "special", just because it is not Latin based.