TTS Pre-processing Issues for Mixed Language Support

Paulseph-John Farrugia*

University of Malta

Abstract. The design of an open domain Text-to-Speech (TTS) system mandates a pre-processing module that is responsible for preparing the input text so that it can be handled in a standard fashion by other TTS components. There are several pre-processing issues that are common across input domains, such as number, date and acronym handling. Others may be specific to the type of input under examination.
In designing a TTS system supporting Maltese for SMS messages in the local context, specific issues are also encountered. In terms of language issues, the practical use of Maltese is heavily characterised by *code-switching* into other languages, most commonly in English. While the resulting language may not be considered 'Maltese' in the strictest sense of the language definition, it creates a state of affairs that cannot simply be ignored by a general purpose system. In respect of the SMS domain, the use of various shorthand notation and lack of phrase structure is encountered.
This paper describes these specific issues in further detail and discusses techniques with which they may be addressed.

1 Introduction

The human reading process (an overview of whose intricacies is given in [1]) is quite a complex one. It is simplicity with which the human mind can easily process even heavily misspelt texts is impressive. Fig. 1 provides an interesting intuitive example of this. Implementing the same kind of flexibility programmatically for a TTS system, however, is not a simple task.

Handling part of this task is the role of the *pre-processing* module. This is the first stage of input processing, organising the input text into a standard format that the following modules can process more easily ([2]). Amongst other things, it is generally responsible for converting numerals and acronyms into their textual interpretation and resolving punctuation. In specialised systems, other tasks may be required of it. For example, in an email to speech system, the preprocessor will be required to remove unnecessary email headers and superfluous formatting. (See [3] for examples.)

In designing a TTS system supporting Maltese for SMS messages in the local context, pre-processing issues specific to this domain are encountered. These will be illustrated in some detail and techniques with which they may be addressed will be discussed.

* This work is supported by MobIsle Communications Ltd. (http://www.go.com.mt)

The following text is taken from an email commonly forwarded around the Internet:

> "*Aoccdrnig to rscheearch at an Elingsh uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, olny taht the frist and lsat ltteres are at the rghit pcleas. The rset can be a toatl mses and you can sitll raed it wouthit a porbelm. Tihs is bcuseae we do not raed ervey lteter by ilstef, but the wrod as a wlohe. Fnnuy how the mnid wroks, eh? ...*"

Irrespective of the validity of its message, it does provide an amusing example of the facility with which the human mind can easily interpret and read out even heavily misspelt texts.

Fig. 1. Reading Misspelled Text

2 Mixed Language Support

In the domain of concern, the specific pre-processing issues arise out of the mixed language nature of the text, the mechanisms used to artificially shorten the messages, orthographic errors and device limitations. The two main concerns, *code switching* and character encoding support, are discussed below. Another pre-processing issue is the handling of *smileys*, such as `;->`, and other abbreviations typical of the SMS domain. This however, can be handled by a simple rewrite process from well-known lists of such common shortcuts.

2.1 Bilinguality and Code Switching

The Maltese Islands are officially bilingual, with both Maltese and English as official languages of the country. English is generally understood as British English, while an official version of Maltese, often termed as *Standard Maltese* (see [4] and [5]) is commonly taught in schools. The latter is distinguished from dialects of Maltese that may be found in certain villages and localities, which differ subtly at various levels of structure (including syntactic, lexical, grammatical and intonational.)

In practice, however, this bilingual situation brings forth a heterogeneous mixture of language use. For various geographical and sociological reasons ([4]), the use of Maltese, whether spoken or written, is often interspersed with words or phrases in English[1], a phenomenon known as *code switching*. Conversely, the local use of English is often marked by certain grammatical and phonetic differences, to the extent that some may want to refer to it as a "Maltese English" dialect ([6]).

For an appreciation of the above, consider, for instance, the following (real-life) SMS text message sample:

> *"jaqaw bdejt tibza tixel il car? xorta qajjimt il qattusa u issa qed tajjat wara il bieb. bring that btl of last time plz qalbi :)"*

[1] And, to a lesser extent, with words in other languages, most typically in Italian.

The extent to which such code switching is acceptable as proper use of the Maltese language (or whether "Maltese English" may be truly considered an English dialect as opposed to a result of a lack of linguistic sophistication) is a matter of near religious debate ([6]). Rather than going into the respective merits of such considerations, a pragmatic approach is preferred whereby this state of affairs is addressed and techniques that can help solve the issues that arise are identified. In this case, a means is required to identify the switching from one language to another at the word level[2].

The intuitive solution would be to resort to a lookup process within extensive dictionaries for the languages involved. This approach by itself, however, is bound to have limited success. Firstly, in an open domain, no dictionary is ever expected to be complete, and hence the resolution for unknown words remains a problem. Secondly, SMS messages often contain a variety of spelling mistakes, rendering a lookup into a classic dictionary prone to failure. However, it is not excluded that the use of dictionary can be a supplementary aid to the language tagging process.

Attention is changed, then, to stochastic approaches for identifying the language of a text, of which two main trends can be identified ([7].) The first involves the use of frequencies of occurrence of short words (such as *u* and *ta* in Maltese) in order to determine the general language of a given corpus. This is not a directly applicable approach here, since the interest lies in identifying the language of origin of each individual word.

A second technique, which has been used for language ([8]) and subject ([9]) classification, is based on *n-gram* probability matching on characters. In essence, for each of a candidate set of languages, an n-gram probability profile is created from a training corpus. Input text is then classified according to the profile that best matches its n-gram probabilities.

An extension is hereby proposed to handle the issue of code switching. Assuming *bigrams* are being used, the procedure can be expressed as follows. Let $\mathbf{L} = \{L_1, L_2, \ldots, L_n\}$ be a set of n candidate languages. For each L_i, let $C_i = \{c_1, c_2, \ldots, c_m\}$ be the set containing those characters belonging to the language L_i, taken from a universal set of characters $\mathbf{C}$.

Given any significantly large corpus known to be entirely (or primarily, since the interest here lies in statistical significance) in a language L_i, it is possible to compute $0 \leq P_{L_i}(a,b) \leq 1$, the probability that the bigram ab occurs in free text written in the language L_i, where $a, b \in \mathbf{C}$. The probability of a word $\mathbf{w} = w_1 w_2 \ldots w_k$, where $w_j \in \mathbf{C}$, belonging to L_i can then be defined as:

$$Pw_{L_i}(\mathbf{w}) = \prod_{j=0}^{k} P_{L_i}(w_j, w_{j+1})$$

[2] Ideally, this should be at the morpheme level, since it is also common to process loan words using Maltese morphological rules. Also, to a certain extent, it may be desirable to phonetize other orthographic elements, such as numerals, in another language, in order to conversely mimic this tendency toward code-switching.

where w_0 and w_{k+1} are defined as the equivalence class of characters in the set $\mathbf{C} \setminus C_i$, that is, those characters not belonging to language L_i (which include spaces and punctuation.) The most likely classification of $\mathbf{w}$, then, is that it belongs to the language L_i that maximizes $Pw_{L_i}(\mathbf{w})$ over all languages in $\mathbf{L}$. In practice, it would also be possible to take the probabilities of the surrounding words for further disambiguating power.

2.2 Character Set Support

Another issue to be tackled arises from the character set required to encode the Maltese alphabet. Proper writing of Maltese mandates the use of certain characters, namely *ċ*, *Ċ*, *ħ*, *Ħ*, *għ*, *Għ*, *ż*, *Ż*, that are not found in the ASCII character set. Electronic input devices—from keyboards to mobile phones—in the local market are usually intended for an English audience (perhaps supporting some other main European languages), and the support for the full Maltese character set ranges from limited (as is the case with most common computer keyboards in use[3], for which specific shortcuts may be required) to non-existent (as in the case of mobile phones.) Given the use of English as a secondary language, this is generally not seen as a major usability issue, and the motivation to have such devices enabled is somewhat limited.

While the advent of Unicode ([10]) is now helping to provide a reference format standard for Maltese text processing (although it still poses some computational issues, as it does not admit the representation of digraphs, such as *ie* and *għ*, as single character units), the situation regarding electronic texts at this point is really split in two. As regards official documents or documents meant for public access, schemes have been utilised to represent the above characters using the ordinary ASCII encoding. The most typical is the use of particular fonts designed for Maltese typesetting (but incompatible with Unicode) whereby the glyphs of the above characters replace the entries for certain punctuation characters ([]). Under other schemes, such as that used in Maltilex ([11]), character escape sequences are used instead.

On the other hand, where the presentation of text is not critical, it is often the case that letters containing diacritical marks are written instead with those ASCII character counterparts that do not, which, corresponding to the above, would be *c*, *C*, *h*, *H*, *gh*, *Gh*, *z* and *Z*. Despite what would constitute intrinsic spelling mistakes, a native speaker can immediately identify and interpret the underlying message[4]. However, such text cannot be directly operated upon prior to resolving the resulting lexical ambiguities.

Resolving this issue is an exercise in spell checking, albeit a simplified one. One could envisage resolving it using a threefold approach (starting from the assumption that the word has already been tagged as a Maltese one.) First, certain re-write rules can be applied. For instance, *c* can be safely re-written as *ċ* in all cases, since the previous character does not exist in the Maltese alphabet.

[3] A Maltese keyboard standard has been defined but is not in widespread use.

[4] Essentially, this is a more natural occurrence of the phenomenon illustrated in Fig. 1.

Secondly, use of a dictionary can sort out some entries, as it will list *żarbun* but not *zarbun*. Finally, one could consider using stochastically based re-write rules trained on a corpus written in two fashions, once in the appropriate encoding and one using the ASCII character set only.

3 Language Classification Implementation

In the following, the process of developing a language classification mechanism for differentiating between Maltese and English words, based upon the ideas from the previous section, is described. Such a mechanism would form part of a more complete pre-processor module.

3.1 Corpora and Bigram Calculation

In order to estimate the n-gram probabilities, suitable corpora for the languages under consideration is required. In order for the probabilities to be meaningful, the corpora are required to be substantially large, and ideally from the same domain as the text that needs to be classified.

Unfortunately, corpora consisting solely of SMS messages already organised as Maltese or English are not readily available, and deriving substantially sized ones would be a very time consuming exercise. An alternative textual corpus is available in the form of the Laws of Malta ([12]). These provide a suitably large and balanced corpora for both Maltese and English. (The fact that the corpora are translations of each other is of peripheral interest here. However, it is useful in providing balance for the frequencies being calculated.)

The laws are available in PDF format, and hence not directly suitable for machine processing. The text contents were thus extracted to plain text files in order that they may be easily processed programmatically. As a result of the extraction process, the files were not completely coherent with respect to the originals, containing some spurious symbols and characters (such as used for formatting or page numbering). However, given that these made up only a very small percentage of the overall corpora, they make negligible effect on the overall frequency calculation results.

The Maltese laws seem to have been encoded with the legacy 'Tornado' fonts which necessitated a mapping from the replaced glyphs to the proper Maltese characters in the Unicode set. The text files were saved in UTF-8 format. Another version of these files was created, where the non-ASCII Maltese characters were replaced by their ASCII counterparts. This was necessary, since, for the reasons explained in the previous section, in the short to medium term, textual input from mobile devices is expected to contain this ASCII version of Maltese. Unless the n-grams are calculated on the ASCII based frequencies, language identification would be expected to show a preference towards English.

An application was written to extract the bigram frequencies from the corpora thus created and save them into an XML file format. For Maltese, the

bigram frequencies were calculated on both the Unicode and ASCII text corpora. These were then used as the basis for a language classification application that tags whitespace separated tokens according to the maximal $Pw_{L_i}(\mathbf{w})$.

3.2 Preliminary Results

The language classifier thus created was used to identify a set of random SMS messages[5] consisting of 1000 whitespace separated tokens (mostly words or numbers, but also containing shortcuts and punctuation which were processed in the same manner.) The results were then hand-checked to verify correctness. With this basic, unpolished, process, a 76% accuracy ratio was obtained.

While being correct three fourths of the time seems a reasonable start, analysing the results one finds that there is room for substantial improvement in the approach. Firstly, given that no attempt at pre-filtering the input was done, some of the cases (such as smileys and punctuation) could not be clearly classified English or Maltese. In this case, the tagging was considered as incorrect, but it is expected that in future revisions a pre-filtering process would take care of these tokens. Secondly, a certain amount of words were actually in languages other than Maltese and English. Again, these were marked as incorrect, even though they contradict the implicit assumption that the input can be accordingly classified.

However, certain misidentification errors do arise out of the limitations of the probabilistic approach taken. One of the notable problems was the word 'I', necessarily an English one, being consistently[6] identified as Maltese. This arises out of the characteristics of corpus chosen for frequency calculation, which would hardly contain any instance of this word, even though in general English it is quite a common one. To solve this, it is necessary to improve the n-gram frequencies by calculating them from corpora that contain this word with a reasonable frequency.

Another prominent issue was the tagging of the word 'u' as Maltese. This word is in fact used both as a shortening of the English word 'you' and also as the proper form of the Maltese 'u' ('and'). In this case, the error is not a result of incorrect training and it is necessary to take the surrounding context into consideration in order to resolve the ambiguity.

3.3 Conclusions and Indications for Further Work

The implementation described above provides a a basic infrastructure for tackling the problem of code switching. However, it is clear that there is still room for further improving the results. One possible improvement would be to further train the language classificator by calculating frequencies on text from the

[5] To preserve anonymity, all header information, including sender number, receiver number and timestamps, were stripped off prior to usage, and only the message text was maintained. Messages were also sequentially randomized, removing any implicit temporal information.

[6] Obviously, this basic approach does not permit otherwise.

domain concerned itself. Creating the corpus could now be simplified by, for instance, using $|Pw_{Maltese}(\mathbf{w}) - Pw_{English}(\mathbf{w})|$ of the tagged text as a confidence level indicator, and consequently manually check only those results which fall below a specific threshold. In this manner, one should expect the results to improve iteratively.

Still, the basic approach fails to account properly for words that can occur in more than one of the languages under consideration. Identifying the situation by itself necessitates a dictionary lookup, but resolving it would require taking the surrounding context into consideration. However, since the surrounding language context is itself statistically inferred, it is possible to imagine a multi-pass approach that resolves ambiguities left open in previous passes. How to best account for context in this manner still remains to be determined.

References

1. Dutoit, T.: An Introduction to Text-To-Speech Synthesis. Volume 3 of Text, Speech and Language Technology. Kluwer Academic Publishers, P.O. Box 322, 3300 AH Dordrecht, The Netherlands (1997)
2. Farrugia, P.J.: Text-to-speech technologies for mobile telephony services, University of Malta, Computer Science Department (2003)
3. Black, A.W., Taylor, P., Caley, R.: The Festival Speech Synthesis System. University of Edinburgh. 1.4 edn. (2002)
4. Borg, A.: Ilsienna. Albert Borg (1988)
5. Borg, A., Azzopardi-Alexander, M.: Maltese. Descriptive Grammars. Routlege, London and New York (1997)
6. Vella, A.: Prosodic Structure and Intonation in Maltese and its Influence on Maltese English. PhD thesis, University of Edinburgh (1995)
7. Grefenstette, G.: Comparing two language identification schemes. In: 3rd International Conference on Statistical Analysis of Textual Data, Rome (1995)
8. Beesley, K.: Language identifier: A computer program for automatic natural-language identification of on-line text. In: Languages at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association. (1988) 47–54
9. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of (SDAIR)-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US (1994) 161–175
10. : Unicode homepage. (http://www.unicode.org)
11. : Maltilex homepage. (http://mlex.cs.um.edu.mt)
12. : Laws of Malta Homepage. (http://justice.gov.mt)