Information Processing and Management 52 (2016) 326–339

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

DeASCIIfication approach to handle diacritics in Turkish information retrieval

Ahmet Arslan*

Computer Engineering Department, Anadolu University, Eskisehir 26555, Turkey

ARTICLE INFO

Article history:
Received 29 July 2014
Revised 21 August 2015
Accepted 28 August 2015
Available online 10 November 2015

Keywords:
Accents
DeASCIIfier
Diacritics restoration
Risk-sensitive evaluation
Stemming
Turkish information retrieval

ABSTRACT

The absence of diacritics in text documents or search queries is a serious problem for Turkish information retrieval because it creates homographic ambiguity. Thus, the inappropriate handling of diacritics reduces the retrieval performance in search engines. A straightforward solution to this problem is to normalize tokens by replacing diacritic characters with their American Standard Code for Information Interchange (ASCII) counterparts. However, this so-called ASCIIfication produces either synthetic words that are not legitimate Turkish words or legitimate words with meanings that are completely different from those of the original words. These non-valid synthetic words cannot be processed by morphological analysis components (such as stemmers or lemmatizers), which expect the input to be valid Turkish words. By contrast, synthetic words are not a problem when no stemmer or a simple first-*n*-characters-stemmer is used in the text analysis pipeline. This difference emphasizes the notion of the *diacritic sensitivity* of stemmers. In this study, we propose and evaluate an alternative solution based on the application of deASCIIfication, which restores accented letters in query terms or text documents. Our risk-sensitive evaluation results showed that the diacritics restoration approach yielded more effective and robust results compared with normalizing tokens to remove diacritics.

1. Introduction

American Standard Code for Information Interchange (ASCII)[1] is a scheme that encodes 128 specified characters into 7-bit binary integers. ASCII contains English alphabet letters (a–z and A–Z), numbers from 0 to 9, and some other special characters. However, a number of languages use characters outside the ASCII range and they have letters with diacritics in their alphabet, such as Czech, Danish, Finnish, French, Greek, Hungarian, Icelandic, Latvian, Lithuanian, Norwegian, Polish, Romanian, Swedish, Spanish, and Turkish. The absence of diacritics in digitally stored text is a severe obstacle to natural language processing (NLP) and information retrieval (IR) for languages with alphabets not covered by the standard ASCII range.

Turkish is an agglutinative and morphologically complex language, where a relatively small set of distinct stems is expanded by rich combinations of derivational and inflectional suffixes to create new meanings.

The Turkish alphabet is a Latin alphabet that comprises 29 letters. It has all the letters of the English alphabet, except "q," "w," and "x." In addition, it has the local characters "ç," "ğ," "ı," "ö," "ş," and "ü" with diacritic symbols, which have been modified from their Latin originals to meet the phonetic requirements (to distinguish different sounds) of the language.

* Tel.: +902223213550.
E-mail address: aarslan2@anadolu.edu.tr

[1] http://tools.ietf.org/search/rfc4949

http://dx.doi.org/10.1016/j.ipm.2015.08.004
0306-4573/

Table 1
List of accented Turkish characters and their ASCII counterparts.

Turkish	ç	ğ	ı	ö	ş	ü	Ç	Ğ	İ	Ö	Ş	Ü
ASCII	c	g	i	o	s	u	C	G	I	O	S	U

Table 2
List of Turkish words and their English meanings.

Turkish word	English meaning	ASCII word	English meaning
kuş	bird	kus	vomit
köyün	your village	koyun	sheep
marş	anthem	mars	planet Mars
çin	China	cin	genie
bakır	copper	bakir	virgin
üçüz	triplet	ucuz	cheap
kılım	my hair	kilim	rug

Diacritics are also referred to as accent marks, which are defined as: "A mark placed above, below, or to the side of a character to alter its phonetic value."[2] Table 1 shows Turkish accented letters and their ASCII equivalents. Due to these non-ASCII letters, Turkish users experience many IR problems on the Internet (Aytaç, 2005).

With regard to accents and diacritics, Manning et al. stated that: "Nevertheless, the important question is usually not prescriptive or linguistic but is a question of how users are likely to write queries for these words. In many cases, users will enter queries for words without diacritics, whether for reasons of speed, laziness, limited software, or habits born of the days when it was hard to use non-ASCII text on many computer systems" (Manning et al., 2008).

However, Turkish texts written in the English alphabet may have different meanings that cannot be distinguished even by a human. For example, an interesting news story by Diaz (2008) described how a Turkish SMS written in English letters resulted in a completely twisted meaning that resulted in homicides.

Turkish users have a tendency to write Turkish search queries without using accented Turkish letters due to the reasons explained above. Therefore, there is a need for an ability to search with and without accents in Turkish IR. For instance, in diacritic insensitive IR, the words *resume* and *résumé* should be treated as if they are the same word.

ASCIIfication, also referred to as *latinization*, is the normalization of tokens to reduce all accented letters to their base character. ASCIIfication is a common practice for achieving accent-insensitive IR; however, this may result in a change in meaning because certain words are distinguished only by their accents. Table 2 shows a list of examples that would retrieve false matches.

Furthermore, the ASCIIfication process yields synthetic words, which can negatively affect downstream morphological analysis components (such as stemmers or lemmatizers) in the processing pipeline. The transformation of the word *hastalığının* into *hastaliginin* is an example. In this case, *hastaliginin* is not a legitimate Turkish word, so it is not recognized by morphological analyzers. In this study, we propose and evaluate an alternative solution based on the application of deASCIIfication for restoring diacritics in query terms or text documents.

The remainder of this paper is organized as follows. Section 2 describes related research. In Section 3, we explain the two deASCIIfication algorithms used in this study. In Section 4, we provide intrinsic evaluation results obtained using the two different deASCIIfication systems. Section 5 describes the experimental setup. In Section 6, we present the experimental results and a discussion. In Section 7, we give robustness results obtained from a comparative risk-sensitive evaluation of different approaches for handling diacritics in Turkish IR. In Section 8, we give our conclusions and suggestions for future research.

2. Related work

2.1. Diacritics restoration

Diacritics restoration (DR) can be defined as the automatic insertion of diacritics into text when they are absent. Due to the continuously increasing volume of user-generated textual content (blogs, forums, wikis, etc.) on the Web, DR tools have become essential components in many important applications, such as IR, machine translation, named entity recognition, corpora acquisition, and the construction of machine-readable dictionaries (Mihalcea, 2002).

DR studies have been performed for many languages, including Arabic (Azmi & Almajed, 2015), Croatian (Šantić et al., 2009), Vietnamese (Do et al., 2013), Romanian (Grozea, 2012), and Turkish; however, the studies in this category are not related to IR.

The first published study of Turkish DR was carried out by Tür (2000). He constructed a character-based language model using an 18 million-word corpus of Turkish and built a hidden Markov model (HMM) whose states denoted Turkish characters-grams and whose state transition probabilities were obtained from the language model. The performance of the deASCIIfier was evaluated using a test data of 8511 words, among which 5864 words needed to be corrected. Tür tested the system using

[2] http://www.unicode.org/glossary/#accent_mark