Original scientific paper

Automatic Diacritics Restoration in Croatian Texts

Nikola Šantić
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
nikola.santic@fer.hr

Jan Šnajder
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
jan.snajder@fer.hr

Bojana Dalbelo Bašić
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
bojana.dalbelo@fer.hr

Summary

The absence of diacritics in digitally encoded text is a common problem for languages whose writing systems are not covered by the standard ASCII character set. It is a deterioration of language in its own right, but also a serious impediment to automated text processing and information retrieval. Restoration of diacritics, if performed manually, is a tedious and time-consuming process. In this paper we describe a robust system for automatic diacritics restoration in Croatian texts. The system combines dictionary look-up and statistical language modelling. Diacritics restoration is evaluated on a corpus of newspaper articles and discussion forum posts. Our experiments show that high levels of accuracy can be achieved with fairly simple and computationally inexpensive methods.

Key words: natural language processing, diacritics restoration, statistical language model, Croatian language

1. Introduction

Verification and correction of spelling errors is probably one of the most used applications of natural language processing. The most common types of errors are orthographic and typing ones. In addition to these, there is another category of spellchecking, which, although not characteristic for English, is relevant to most other European languages – restoration of diacritics. Diacritics can be found in, amongst others, Spanish, Portuguese, French, German, and most of the Scandinavian and Slavic languages. Automated restoration of diacritics is useful not only for the restoration of legacy texts that were typeset without diacritics, but also for ever-larger amounts of contemporary content in which the diacritics are absent. The main reason for this is the lack of an accepted encoding standard, so the users find it simpler to leave out the diacritics while typing. This phenomenon is especially common in casual forms of electronic communications such as e-mail, discussion forum posts, and chats. In fact, most texts found on Internet blogs and discussion forums are in “mixed” form, partly with diacritics and partly without. Another common cause of diacritics’ absence is conversion to formats of non-compatible encodings, for instance, when text is extracted from a PDF document. Although a minor problem for a human reader, the absence of diacritics is a major setback for automated text processing and information retrieval.

The problem of diacritics restoration, although specific to each language, has given rise to two basic approaches: *word-based* and *character-based* restoration (Mihalcea, 2002). Word-based approaches are commonly knowledge-intensive systems that rely on dictionaries and statistical language models, which makes these systems language dependant. They require a large source of grammatically correct text in order to build a useful model, and need more processing time because of the pre-processing (e.g., tokenization, tagging, etc.). On the other hand, character-based systems rely on language-independent algorithms that use statistical information gained from the training data. They are much simpler, faster, easier to implement, and do not require any language-specific resources. However, in languages where the change of diacritics has a grammatical or semantic role, word-based systems are much more reliable (Tufiş, 2007). Only in languages where the diacritics can be restored without examining the context can the character-based systems be expected to yield high accuracy. Therefore, when choosing an approach, different aspects should be considered: the role of diacritics in the language, availability of adequate training data, required processing speed, and users’ requests and needs.

In this paper, we describe and evaluate a word-based diacritics restoration system designed for Croatian language. The system is based on word recognition and selection using a dictionary and a statistical language model. To the best of our knowledge, this is the first work that directly addresses the issue of diacritics restoration for Croatian language.

The rest of the paper is structured as follows. A brief overview of related work is given in the next section. Section 3 discusses the peculiarities of diacritics restoration for Croatian language. In Section 4 we describe our diacritics restoration system, while in Section 5 we present the experimental evaluation. Section 6 concludes the paper.

2. Related work

Due to orthographic differences among languages, it is impossible to define a universal solution to diacritics restoration. Still, some generic ideas can be adjusted or refined to fulfil the requirements of a particular language.

Spriet and El-Bèze (1997) use part-of-speech tagging to restore accents in French. Their method tags each word with its type and its relation to other words, based on the context and statistical n-gram information. Unknown words are ignored, since the majority of unknown words need not be restored anyway. They evaluate the method on a 19,000 word corpus, obtaining a rather satisfactory accuracy of 99.31%.

Pauw et al. (2007) have shown that for resource-scarce languages, character-based approaches are more successful than word-based approaches. Using machine learning methods, they managed to restore diacritics of various African languages with an accuracy of up to 70%.

Mihalcea (2002) presents a method based on a genetic algorithm applied on the letter level. In the restoration of a Romanian electronic dictionary, she reports letter-level accuracies of over 99%.

DIAC$^+$ (Tufiş, 2007) is an advanced diacritics restoration tool – originally developed for Romanian language – that uses both word and character based approaches. DIAC$^+$ uses a tagging system similar to that of El-Bèze et al. (1994) and three dictionaries: a dictionary of words with diacritics, a dictionary of words with diacritics stripped off, and a list of words currently being processed but not present in either of the other two dictionaries. Restoration candidates are first chosen from one of the three dictionaries and then morphosyntactically tagged. In case of ambiguity, either the user is queried or the restoration proceeds automatically according to chosen probabilistic parameters. For unknown words, a character-based n-gram model is used. Evaluated on a 118,000 words corpus, DIAC$^+$ achieves accuracy of almost 99% on pre-tagged text and 97% when the text is untagged.

The only published work on diacritics restoration related to Croatian language is (Scheyt et al., 1998). The method described there was developed as a part of the *Serbo-Croatian dictation and broadcast news speech recogniser*. Diacritics restoration was required since their absence from most Internet sites prevented further data processing. The proposed method marks as correct all words found in the dictionary, ignoring possible ambiguity. Unrecognised words are assigned to the nearest neighbour based on string similarity. In the last step, a letter trigram model is generated and used to score likelihood of the different possible character sequences for the remaining words. Accuracy of 95% is reported, a fairly high result considering the simplicity of the procedure.

All the above mentioned examples show that good results can be achieved even with the most rudimentary approaches, but also that the properties of the language still play a major role.

3. Diacritics in Croatian

There are five diacritical characters in Croatian: *č*, *ć*, *š*, *ž* and *đ* (a diacritic is also contained in the digraph *dž*). As most other diacritical characters, none of these are present in the standard ASCII character set, the first widely used encoding standard. First code page expansions to start including diacritics were developed during the 1980s – the ISO 8859 family. Croatian language was supported there within both the Central European language family (Latin-2 encoding) and the Southeast European family (Latin-10 encoding). Different operating systems also developed their own encoding standards, and the Croatian diacritical characters were included both in windows-1250 and in Macintosh Southeast code pages. Unfortunately, all the available encodings were mutually incompatible. First unifying standards came with the advent of Internet. Today, the two most prevalent are UCS and Unicode. Both use up to 32 bits per character, which allows the encoding of all characters of all known languages. A lot of effort has been made to synchronize and translate one standard into the other. Despite this, a unified standard still does not exist – for example, the three most popular Croatian news portals still use different encodings: UTF-8, windows-1250, and ISO-8859-2.

Substitution of diacritics

Although the above-mentioned encoding standards represent an improvement from the times when diacritics were not supported at all, many users still carry a habit of omitting the so-called “Croatian characters”, and substituting them with conventionalised variants. Table 1 contains some of the most common substitutes for each diacritical character.

Table 1: Most common diacritic substitutions

Diacritical character	Substitute characters
č, ć	c, cc, ch, cx, cy
š	s, ss, sh, sx, sy
ž	z, zz, zh, zx, zy
đ	d, dj, dy

None of the above-mentioned substitutions are flawless. The most frequent substitution scheme is the omission of diacritics (*č* and *ć* become *c*, *š* becomes *s*, *ž* becomes *z*, and *đ* becomes *d*). This scheme often causes ambiguity, e.g., between *kuca* (knocks) and *kuća* (house), *kos* (blackbird) and *koš* (basket), *zao* (evil) and *žao* (sorry), *voda* (water) and *vođa* (leader). Fortunately, most words have only one or two valid diacritical forms, while three valid diacritical forms are extremely rare, e.g., *obuci* (wear), *obuči* (train), and *obući* (to footwear). Switching to a substitution scheme that adds an extra character eliminates this ambiguity, but it also makes words less readable, e.g., *rječca* (small word) becomes *rjeccca*, *čišći* (cleaner) becomes *cxisxcxi,* etc.

In what follows, words that contain a potential substitution for a diacritic (characters *c, s, z,* or *d*) will be referred to as *C-words*. C-words are words that are to be considered for restoration (this is because all substitutions schemes use one of these four letters as the substitute character). Words that contain a diacritical character will be referred to as *D-words*. Note that a word can be both a C-word and a D-word, e.g., *cvijeće* (flowers). A *C-word variant* is obtained by applying (inverse) substitutions on a given C-word; a complete set of C-word variants of a given C-word is obtained by permuting all possible substitutions. For example, C-word variants of *staza* are *staza, staža, štaza,* and *štaža,* of which only the first two are valid words.

Statistical analysis

Difficulty in selecting a correct word form depends greatly on the properties of the language itself. Table 2 shows statistical data for Croatian language gathered from three different samples: newspaper articles with correct diacritics (*valid*), discussion forum posts with both diacritics and substitutions (*mixed*), and newspaper articles and discussion forum posts from which the diacritics were removed (*removed*). The evaluated texts were of various sizes, ranging from 10,000 to 100,000 word tokens. The C-word and D-word ratios are calculated on the word level, while for substitute characters and diacritics ratios this is done on the letter level.

Table 2: Statistical analysis of diacritics on three samples

	Samples		
	Valid	**Mixed**	**Removed**
C-word ratio	45.7%	48.5%	53.8%
D-word ratio	16.3%	10.1%	–
Substitute characters ratio	10.5%	12.2%	14.1%
Diacritics ratio	3.2%	1.3%	–

The C-word ratio on valid text sample suggests that more than half the words can be marked off as correct without any further processing. Also, the D-word ratio on the same sample suggests that only every sixth word actually contains a diacritic, which means that a text in which the diacritics are absent is already more than 80% correct. As expected, compared to the diacritically valid text, in forum texts (*mixed*) the proportion of C-words is higher whereas the proportion of D-words is lower.

Another interesting parameter of a C-word is the number of its valid C-word variants. C-word variants are considered valid if they are present in the dictionary. Based on a sample of texts containing up to 100,000 words with diacritics removed, the following results were obtained: 88.6% of the C-words have a single valid variant, 7.4% have two, and only 0.1% have three valid variants. The rest (3.9%) are words not present in the dictionary, either misspelled or uncommon. These results imply that most of the C-words can be unambiguously restored using a wide-coverage dictionary. Still, for the process to be fully automated, a method to restore the ambiguous cases is required.

4. System description

The diacritics restoration system described in this work uses a dictionary backed up by a statistical language model. The restoration procedure is divided into three successive steps: tokenization, candidates generation, and the selection of the correct word form. The architecture of the system is shown in Figure 1.

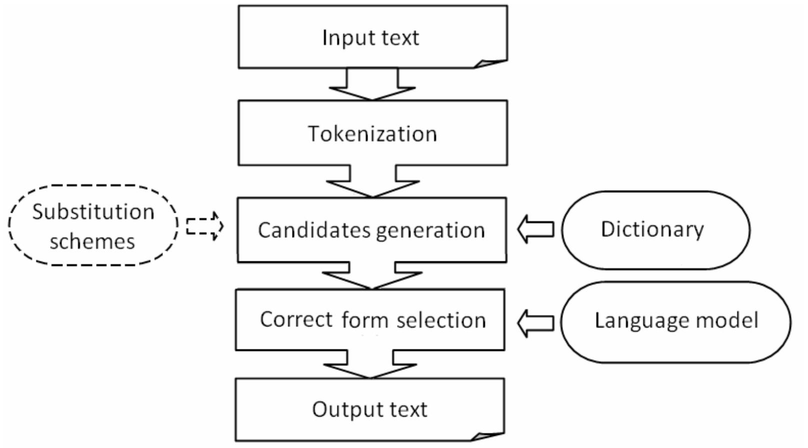

Figure 1: Architecture of the diacritics restoration system

Tokenization

Diacritics restoration in this system is word-based, meaning the lines of input text first have to be tokenised into word tokens. The line is split on whitespace and punctuation characters. Word-based diacritics restoration relies on a context of each word, i.e., a sequence of words that surround it. Our system uses a left and a right context – words that precede and follow a given word – with an adjustable size of the context window. The system processes the text one line at a time, even if the lines contain multiple sentences; preliminary evaluation has shown that, even if the context extends beyond the sentence boundaries, the results do not deteriorate in any way.

Candidates generation

In this step all C-word variants of a word currently being restored are generated. If the word is not a C-word, it is marked off as correct; as shown in the previous section, this eliminates more than half of the words. For the remaining words all their C-word variants are generated. The number of variants grows exponentially with the number of substitute characters. Since every character can be substituted by either two or three characters (*s* : {*s,š*}; *c* : {*c,č,ć*}), the base of the exponential function takes on values between 2 and 3. In practice, a C-word has on average 1.1 substitution characters in valid texts and 1.2 in texts with removed diacritics. These numbers are small enough to avoid combinatorial explosion.

The input text is processed from left to right, token by token, so the left context of the word will have been processed by the time we create the C-word's variants. However, the right context will not be processed yet, so the C-word variants of each token from the right context need to be created. Ultimately, a Cartesian product of generated variants sets is used; its size increases exponentially with the size of the right context.

Most of the C-word variants generated in this step are not valid words, so they have to be checked against a dictionary. The dictionary used in this system is derived from an automatically acquired inflectional lexicon totaling over 3.5 million words (Šnajder et al. 2008). Since all the words looked up in the dictionary will be C-word variants, the dictionary needs to contain only C-words and D-words; this reduces the size of the dictionary to 750.000 entries.

Selection of the correct word form

As shown previously, using dictionary alone will correctly and unambiguously restore the majority of C-words (more than 88% on our sample). We consider this performance as the baseline. The remaining 12% of the C-words have ambiguous variants, e.g., *suma* being either *suma* (a sum) or šuma (a forest). In such cases, the word's context is examined using a language model.

The language model is a statistical model of word sequences (Jurafsky, 2000). It contains data on probabilities of specific word combinations to appear in text. In our case, the input to the language model is a word sequence w_1^n consisting of the C-word and its left and right contexts. The output of the language model is the probability $P(w_1^n)$ of the given sequence. The probabilities of different sequences are compared and the most likely one is chosen as the correct one. For practical reasons (memory consumption and training data availability), in a language model a sequence probability $P(w_1^n)$ is approximated by n-gram probabilities. In this work, we chose to use a bigram language model, thus the probability of word sequence $P(w_1^n)$ is approximated by a product of conditional probabilities:

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_1^{k-1}) \approx \prod_{k=1}^{n} P(w_k \mid w_{k-1}) \ .$$

In other words, the probability of a word in a sequence is calculated based only on a single word preceding it. It has been shown that this simplification still produces satisfactory results.

The main drawback of the above-described model is in the limited size of the training corpus: no matter how large the corpus, there will always exist bigrams that are not present in the training corpus but perfectly acceptable in the language. Such bigrams would be assigned a zero probability, and consequently the whole word sequence would be discarded. This problem is commonly addressed by *smoothing*, the technique of assigning non-zero probabilities to "zero probability n-grams". The system described here uses the Witten-Bell discounting method (Witten & Bell, 1991), treating the zero-frequency bigrams as events that did not yet occur. Their probability can be calculated by counting the number of times a new bigram appears in the corpus. This number is, in turn, equal to the number of bigram types – different bigrams that appear in the corpus. Thus, if bigram $w_{n-1}w_n$ is not contained in the corpus, in order to calculate the probability $P(w_{n-1}w_n)$, we use the number of bigram types starting with w_{n-1}. Let T be this number, Z the number of zero-probability bigrams, and N the total number of bigrams starting with w_{n-1}. The smoothed probability P^* of word w_i conditioned on a preceding word w_{i-1} equals to:

$$P^*(w_i \mid w_{i-1}) = \frac{T(w_{i-1})}{Z(w_{i-1})(N(w_{i-1}) + T(w_{i-1}))}.$$

This extra probability must be discounted from the probability of all the seen bigrams, using the following equation for the total probability mass where C is the total count of the given word sequence:

$$\sum_{i:C(w_x w_i)>0} P^*(w_i \mid w_x) = \frac{C(w_x w_i)}{C(w_x) + T(w_x)}.$$

Smoothing is a relatively simple concept, but one that can significantly improve the results, especially when training corpora is small.

5. Evaluation

For the purposes of evaluation, two different corpora were used: a ca. 100,000 word corpus of newspaper articles collected from the Internet site of *Glas Slavonije*[1] and a ca. 30,000 word corpus of discussion forum posts collected from the *forum.hr* web site. Diacritical characters were removed from the newspaper corpus and replaced with substitute characters. The substitution scheme used was the omission of diacritics because this scheme causes the most ambiguity. The forum corpus was not altered. These two corpora were used to evaluate different system configurations by comparing the original corpus to the one automatically restored (for newspaper corpus) or comparing manually and automatically restored text (for forum corpus). The context window extended one word to the left and one word to the right; in our preliminary experiments, this showed to yield the best results. The accuracy of diacritics restoration on the two corpora is shown in Table 3; performance improvement over the dictionary baseline is shown in parentheses.

[1] http://www.glas-slavonije.hr

Table 3: Accuracy of diacritics restoration on different corpora

	Newspaper articles	**Forum posts**
Unrestored	80.72%	92.00%
Dictionary only (baseline)	97.07%	97.95%
Dictionary + Language model (unsmoothed)	97.65% (+ 0.6%)	98.38% (+0.4%)
Dictionary + Language model (WB smoothing)	98.81% (+1.8%)	99.35% (+1.4%)

As shown previously, unrestored texts are already over 80% correct, and using only a dictionary-lookup restores the majority of unambiguous words. The improvement over the dictionary baseline is achieved using the bigram language model built from a 2.5 million word corpus of newspaper articles and literary works. Using the same model with smoothing results in restoration accuracy of up to 99%. In comparison to diacritics restoration systems for other languages, this is a very satisfying accuracy, especially considering the computational simplicity of the method.

The incorrectly restored words are of various types, but in each of the corpora a different type prevails. In the newspaper corpus, most errors are made on location and person named entities. This problem cannot be solved using the dictionary nor the language model due to limited corpus coverage – one possible solution would be for the user to define the corresponding substitutions explicitly. Most mistakes made in the forum corpus are misspellings not recognizable by the dictionary. This could be remedied by adding common spelling mistakes to the dictionary or by approximate matching. This can, however, increase the number of errors on diacritically valid words.

Although the language model does not contribute to the performance as much as the dictionary does, it is nevertheless required to achieve peak performance. This is evident by examining the most frequent restoration errors in the corpus (Table 4). The use of the language model reduces the frequency of these errors by more than 90%. This is a substantial improvement, especially if the text is meant for further automatic processing.

Table 4: Most frequent restoration errors in the newspaper articles corpora

Ambiguous C-word variants	**Error count**	
	Dictionary	**Dictionary + Language model (smoothed)**
posto – pošto	74	6
gdje – gđe	69	2
nas – naš	47	3
grada – građa	44	2
posao – pošao	30	1

6. Conclusion

Restoration of diacritics is relevant for most European languages. It is especially important in view of the fact that – despite the efforts to define a unifying encoding standard – a significant amount of diacritically invalid text is still being generated on a daily basis. Such texts present a problem for automated text processing and information retrieval.

In this work we have presented a robust word-based system for diacritics restoration in Croatian texts. The system relies on a dictionary and a bigram language model, it does not require any pre-processing, and is computationally inexpensive. Evaluation shows that good results can even be achieved using only dictionary-lookup, while the use of the language model increases restoration accuracy to almost 99%. The described system will soon be available on-line.[2]

For future work, the performance of the system could be further improved by using a user-defined list of unknown words and dynamic loading of corpus-specific language models.

Acknowledgements

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia and under the Grant 036-1300646-1986.

References

Bèze, M.; Mérialdo, B.; Rozeron, B.; Serouault, A., M. Accentuation automatique de texte par des méthodes probabilistes. // *Technique et sciances informatiques*. 13 (1994), 6; 797–815

De Pauw , G.; Wagacha, P.W.; de Schryver G. Automatic diacritic restoration for resource-scarce languages. // *Lecture Notes in Computer Science*. 4629 (2007); 170–179

Jurafsky, D.; Martin, J.H.; Kehler, A.; Vander Linden, K.; Ward, N. An introduction to natural language processing, computational linguistics, and speech recognition. MIT Press, 2000

Mihalcea, R.F. Diacritics restoration: Learning from letters versus learning from words. // *Lecture notes in computer science*, 2276 (2002); 96–113

Scheytt, P.; Geutner, P.; Waibel, A. Serbo-Croatian LVCSR on the dictation and broadcast news domain. // *IEEE international conference on acoustics, speech and signal processing*, *Volume 2*. 1998, 897-900

Šnajder, J.; Dalbelo-Bašić, B.; Tadić, M.. Automatic acquisition of inflectional lexica for morphological normalisation. // *Information Processing and Management*, 44 (2008), 5; 1720–1731

Tufiş, D.; Ceauşu, A. DIAC$^+$: A professional diacritics recovering system. // *6th language resources and evaluation conference*. ELRA, 2007, 167–174

Witten, I. H.; Bell, T. C. Estimating the probabilities of novel events in adaptive text compression. // *IEEE transactions on information theory*. 37 (1991), 4; 1085–1094

[2] http://ktlab.fer.hr/diacro