A Pointwise Approach for Vietnamese Diacritics Restoration

Tuan Anh Luu

Kazuhide Yamamoto

Department of Electrical Engineering, Nagaoka University of Technology

{anh,yamamoto}@jnlp.org

***Abstract*—The automatic insertion of diacritics in electronic texts is necessary for a number of languages, including French, Romanian, Croatian, Sindhi, Vietnamese, etc. When diacritics are removed from a word and the resulting string of characters is not a word, it is easy to recover the diacritics. However, sometimes the resulting string is also a word, possibly with different grammatical properties or a different meaning, and this makes recovery of the missing diacritics a difficult task for software as well as for human readers. This paper is the first to study automatic diacritic restoration in Vietnamese texts. Modern Vietnamese is a complex language with many diacritical marks, and white space does not always function as a word separator. This paper proposes a pointwise approach for automatically recovering missing diacritics, using three features for classification: *n*-grams of syllables, *n*-grams of syllable types, and dictionary word features. Our experiments show that the proposed method can recover diacritics with a 94.7% accuracy rate.**

Keywords- Vietnamese, automatic diacritic restoration, pointwise approach, natural language processing, classification.

I. INTRODUCTION

Spell checking, which involves detecting and correcting spelling errors, is one of the most common natural language processing applications. The most frequent errors are orthographic and typing errors. However, there is an additional category of spell checking that is needed for most European languages (although not for English) and for some African and Asian languages: the restoration of diacritics. Automated restoration of diacritics is useful for reconstructing legacy texts that were typeset without diacritics. In addition, it is needed for a growing number of contemporary texts that lack diacritics, mainly because there is no accepted standard for encoding diacritics and users therefore find it easier to omit them when they type. This practice is especially common in casual forms of electronic communication such as e-mails, posts on discussion forums, and chats. Thus, missing diacritics pose a serious problem not only for automatic text processing and information retrieval, but also for human readers.

There are two basic approaches to diacritic restoration: word-based and character-based [8]. Word-based approaches are usually implemented as knowledge-intensive systems that rely on dictionaries and statistical language models and are therefore language dependent. These approaches require large corpuses of grammatically correct text in order to build useful models, and they require considerable preprocessing time for tokenization, tagging, and other tasks. In contrast, character-based systems use language-independent algorithms based on statistical information that has been learned from training data. For languages in which diacritics signal grammatical or semantic roles, word-based systems are much more reliable than character-based systems [3]. In general, the choice between the two approaches for restoring diacritics will depend on several factors: the role of diacritics in the targeted language, the availability of adequate training data, the processing speed that is required, and user requests and needs.

Like European languages, modern Vietnamese uses the Latin alphabet. However, in addition to the characters used in English, Vietnamese has letters that are modified with diacritics: ***đ***, ***ă***, ***â***, ***ê***, ***ô***, ***ơ***, and ***ư***; and it is necessary to use an Input Method Editor (IME) to enter these special characters in electronic texts. However, IMEs are slow, difficult to install and use. Therefore, many Vietnamese choose to use non-diacritical Vietnamese, which can be entered using any computer and is easier and quicker to type. However, non-diacritical Vietnamese is difficult to understand and can be very confusing.

Word-based approaches to Vietnamese language processing face two major challenges: there are not enough textual data (such as dictionaries and corpuses), and the Vietnamese language does not have a word separator (this is a problem because word-based approaches must preprocess word segmentations). Phuong (2007) [10] reported a 97% accuracy rate for word segmentation of diacritical Vietnamese. However, the accuracy rate will be considerably lower for non-diacritical Vietnamese, where word segmentation is much more complex and difficult.

The abundance of diacritics along with the absence of a word separator also make Vietnamese a difficult language for traditional character-based restoration, which can be expected to yield a high degree of accuracy only for languages whose diacritics can be restored without examining the context [7]. A new and powerful approach to restoring diacritics is therefore needed for the Vietnamese language.

This is the first research to address the problem of restoring diacritical marks in non-diacritical Vietnamese texts. We propose a pointwise approach that automatically restores missing diacritics using three types of features for classification: *n*-grams of syllables, *n*-grams of syllable types, and dictionary word features. The pointwise approach is simple, powerful, and relatively robust to rare cases that may occur in the text, with little reduction in accuracy [1].

The rest of this paper is organized as follows. Section II presents a brief overview of Vietnamese orthography and some statistical data. Section III describes our approach to restoring diacritics, and Section IV shows the

978-0-7695-4886-9/12 $26.00 © 2012 IEEE
DOI 10.1109/IALP.2012.18

results of an experimental evaluation of our approach. Finally, Section V presents our conclusions.

II. Vietnamese Orthography – Some Statistical Data

Modern written Vietnamese uses the Latin alphabet, known as "quoc ngu" (*quốc ngữ*) in Vietnamese. Quoc ngu consists of 29 letters:

- The 26 letters of the English alphabet except *f*, *j*, *w*, and *z* (which are, however, found in imported loanwords), and
- Seven letters that are modified with diacritics: *đ*, *ă*, *â*, *ê*, *ô*, *ơ*, and *ư* (along with their uppercase variants).

In addition, diacritics are used to indicate different tones in Vietnamese. The tone markings (using the letter ***a*** as an example) are ***a*** (no mark), ***à***, ***á***, ***ả***, ***ã***, and ***ạ***. Letters modified with diacritics and tone markings can be combined to produce complex diacritics such as ***ằ***, ***ở***, ***ể***, and ***ẫ***.

Unfortunately, the Vietnamese language has a problem similar to one found in the Chinese and Japanese languages, where white space is not used to separate consecutive words. In Vietnamese, spaces may be used to separate syllables *within* words, so that the white space does not always serve as a word separator.

In non-diacritical Vietnamese, diacritical marks are removed from modified characters, leaving only the Latin alphabet characters. For example, the characters *"a", "à", "á", "ả"* and *"ạ"* all become *"a"* when the diacritical marks are removed. Approximately 85% of the syllables in Vietnamese and 95% of the words contain letters that use diacritics. All of the letters that involve diacritics are shown in Table I:

TABLE I. Rules for Changing Vietnamese to Non-diacritical Vietnamese

Diacritical Vietnamese Character	***Non-diacritical-character***
"a", "à", "ả", "ã", "á", "ạ", "ă", "ằ", "ẳ", "ẵ", "ắ", "ặ", "â", "ầ", "ẩ", "ẫ", "ấ", "ậ"	"a"
"e", "è", "ẻ", "ẽ", "é", "ẹ", "ê", "ề", "ể", "ễ", "ế", "ệ"	"e"
"i", "ì", "ỉ", "ĩ", "í", "ị"	"i"
"o", "ò", "ỏ", "õ", "ó", "ọ", "ơ", "ờ", "ở", "ỡ", "ớ", "ợ", "ô", "ồ", "ổ", "ỗ", "ố", "ộ"	"o"
"u", "ù", "ủ", "ũ", "ú", "ụ", "ư", "ừ", "ử", "ữ", "ứ", "ự"	"u"
"y", "ỳ", "ỷ", "ỹ", "ý", "ỵ"	"y"
"đ", "d"	"d"

As can be seen in Table I, it is simple to transform diacritical Vietnamese to non-diacritical Vietnamese, but the inverse transformation is complicated. For example, the syllable "*chờ*" (wait) in diacritical Vietnamese becomes *"cho"* in non-diacritical Vietnamese. However, the string *"cho"* in non-diacritical Vietnamese can stand for any of the words "*cho*" (give), "*chó*" (dog), "chợ" (market), "*chờ*" (wait), and so on. In what follows, we refer to strings that are like *"cho"* insofar as they can stand for more than one diacritical or non-diacritical syllable as "ambiguous stripped syllables".

Words in non-diacritical Vietnamese are constructed from 1525 strings (syllables), and diacritics can only be restored for approximately 20% of these strings (such as "*nghich*" and "*hin*") without examining the context. We call such strings "unambiguous stripped syllables".

According to Simard (1998) [5], approximately 85% of the words in arbitrary French texts contain no accents, while more than 50% of the remaining words can be deduced deterministically from the unaccented form. Consequently, in the case of French, with the use of a good dictionary, accents can be restored to an unaccented text with a success rate of nearly 95% .

Vietnamese uses diacritical marks more extensively than other languages such as French, Romanian, and Croatian, and when they are missing it is much more difficult to restore them.

III. Pointwise Approach

In other languages, the problem of restoring diacritics has been solved by various researchers using word-based approaches, but the fact that white space does not always separate words in Vietnamese poses difficulties for such approaches, as they require word segmentation preprocessing. Therefore, in this paper, we propose and evaluate a syllable-based method for restoring diacritics in Vietnamese.

A. Diacritic Restoration Problem and Sequence Labeling

In machine learning, *sequence labeling* is a pattern recognition task that involves the algorithmic assignment of a *category label* to each member of a sequence of observed values. *Part-of-speech tagging*, in which a part of a speech must be assigned to each word in an input sentence or document, is a common example of a sequence labeling task.

If we view a sentence as a sequence of syllables, syllables as objects, and diacritics as classes, diacritic restoration becomes the task of assigning each syllable in a sentence to only one diacritic class. In other words, diacritic restoration becomes a sequence labeling problem, as depicted in Fig. 1.

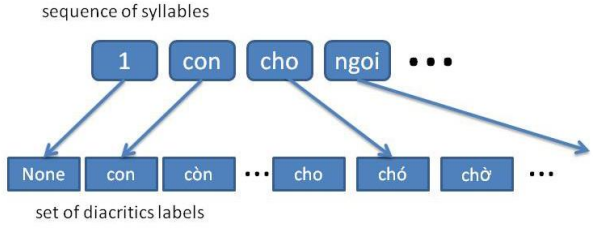


Fig. 1. Sequence labeling problem with syllables as objects and diacritics as classes

There are two basic approaches to sequence labeling: it is done either with or without reference to neighboring labels.

Approaches that examine neighboring labels include hidden Markov models (HMMs) and conditional random fields (CRFs). However, these methods are not effective

when new words (unknown words) are encountered. To use these methods for Vietnamese, it would be necessary to construct a very large dictionary to reduce the number of unknown words and to optimize the speed and accuracy of the methods. Currently, the largest Vietnamese dictionary contains approximately 40,000 words, which is an insufficient number.

In contrast, a pointwise approach does not refer to neighboring labels; it treats sequence labeling as a set of *independent* classification tasks, one for each member of the sequence. The pointwise approach assumes that every decision about a syllable's diacritic is independent of decisions about neighboring syllables.

In general, the problem of classification can be characterized as follows. We have a training set of syllables, each labeled with one class or more classes, which we encode via a data representation model. Typically each syllable in the training set is represented in the form $(\vec{x}, c)$, where $\vec{x} \in R^n$ is a vector of features, and c is the class label. Next, a model class and a training procedure are specified. The model class is a parameterized family of classifiers from which the training procedure selects one classifier.

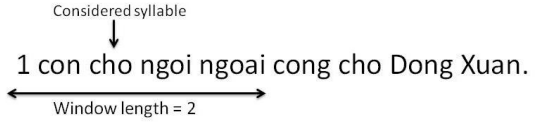
B. *Pointwise Approach to the Diacritic Restoration Problem*

The pointwise approach focuses on one syllable of a sentence, using information about the syllable contained in the data representation: the feature vector and the diacritic class label described in the previous section.

For example, given a non diacritical sentence **s** = $s_1 s_2 \ldots s_n$ (s_i is syllable with $1 \leq i \leq n$) as input. The diacritic restoration whether a syllable s_i can depend on any number of features based on the surrounding non diacritical syllables. We used three types of features that are commonly used in pointwise approaches: *n*-grams of syllables, *n*-grams of syllable types, and dictionary word features:

- **N-grams** of syllables indicate which syllables surround the given syllable. A window size W is specified, and only syllables within this window are used in the classifier's analysis (see Fig. 2). Approximately 70% of the words in the Vietnamese language are composed of two syllables, and approximately 14% are composed of at least three syllables. The high frequency of two-syllable compounds suggested using the window sizes W = 2 and W = 3.
- **N-grams of syllable types** use the following symbols to characterize surrounding syllables:
  - Upper case syllable (U): syllable begins with upper case letters.
  - Lower case syllable (L): syllable contains only lower case letters.
  - Number (N) syllable is a number.
  - Other (O): syllable is something else (other), such as a symbol.
- **Dictionary word features** are dictionary words that contain the given syllable (see Fig. 3)

For example, consider the first occurrence of *"cho"* in *"1 con cho ngoi ngoai cong cho Dong Xuan"* (a dog sitting next to Dong Xuan market's port). Figures 2 and 3 show the information about the surrounding syllables that is stored in the feature vector for use by the classifier.



	Syllable	syllable type
1-gram	"1", "con", "cho", "ngoi", "ngoai"	"N", "L"
2-gram	"1 con", "con cho", "cho ngoi", "ngoi ngoai"	"NL", "LL"

Fig. 2. Syllable and syllable type 1-gram and 2-gram features with window length of 2.

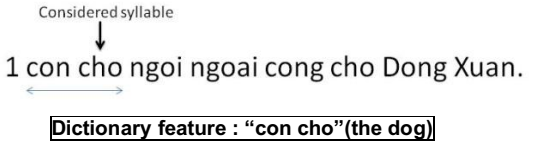


Fig. 3. Dictionary word features

In this example, the first occurrence of the string *"cho"* is represented by the feature vector *("1", "con" , "cho", "ngoi", "ngoai", "1 con", "con cho", "cho ngoi", "ngoi ngoai", "N", "L", "NL", "LL", "con cho(dictionary)")*.

The second string "*cho*" is represented by the vector *("ngoai", "cong", "cho", "Dong", "Xuan", "ngoai cong", "cong cho", "cho Dong", "Dong Xuan", "L", "U", "LL", "LU", "UU", "cong cho(dictionary)")*.

These feature vectors are created for each syllable, and a classifier then determines the diacritical marks.

IV. EXPERIMENTS

As in English, proper names in Vietnamese begin with capital letters. There are no rules for these syllables, nor do they have any relation to the context, so only lower case syllables were provided with diacritical marks.

We wrote a simple crawler that methodically scans through journalism pages to create a corpus of the text it's looking for. We then removed all diacritical marks, using the rules given in Table I. We divided the corpus into two parts: one for training and one for evaluation.

A. *Baseline Methods*

We used two simple baseline methods:

- Random choice: uses a random function to choose diacritical marks from a dictionary for the syllable, and
- Most frequent syllable: always assigns the most probable syllable found in the training data.

We performed experiments on a sample 15 Mb text that contained approximately 1,100,000 syllables. Table II shows the results for the two baseline methods and the pointwise approach.

TABLE II. RESULTS FOR TWO BASELINE METHODS AND THE POINTWISE APPROACH

	Random choice	***Most frequent***	***Pointwise approach***
Accuracy	15.9%	71.8%	94.7

As noted earlier, when a good dictionary is used, accents can be restored to an unaccented French text with a success rate of nearly 95% [5], and a success rate of 97% can be achieved for a Croatian text [7][9]. Diacritic restoration in Vietnamese is much more difficult than in French or Croatian.

B. Pointwise Approach

At the beginning of our experiments, we built a classifier for all syllables. However, because Vietnamese has many syllables and is quite complex, the success rate was not good. We therefore built a classifier for each syllable.

We used a linear support vector machine (SVM), implemented in the LIBLINEAR software package [2], to solve the classification task. SVMs are well suited for this task, as it focuses on the given syllable regardless of any increase in the number of outliers that may arise from difficult or rare cases occurring in the training data. Many machine-learning methods are heavily influenced by outliers, and this reduces their accuracy for more common cases. In contrast, SVMs are relatively robust to the occurrence of rare cases, so that even when these are present, there is only a small reduction in accuracy.

With the chosen features, the accuracy of pointwise approach depends on the window length W and the size of the training data. The graph in Fig. 4 shows the results of our experiments with W = 2 and W = 3 and training data sizes of 10, 20, 40, 80, 160, and 320 Mb.

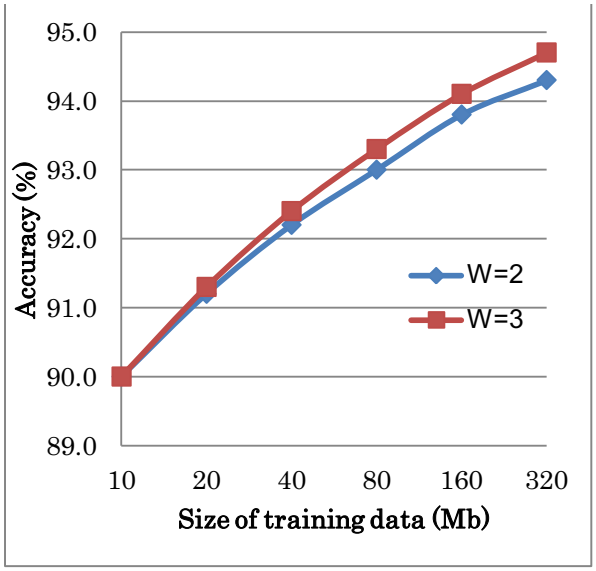


Fig. 4. Results for the proposed pointwise approach

Both the training data and the test sets contain journalism text. Because journalism text contains many unknown words and errors, the accuracy of diacritic restoration is always lowest in the case of these texts [3], [7].

Figure 4 shows that the results in every case are better when the window size W is 3 rather than 2. The curve in the graph is almost linear, and the accuracy improves as the size of the training data set increases.

The highest accuracy obtained was 94.7%. For a complex language such as Vietnamese that uses so many diacritical marks, this is an acceptable result.

V. CONCLUSION

The automatic insertion of diacritics into written Vietnamese texts is important for many applications, including search engines, text-to-speech engines, speech translation, mobile message-reading, and talking dictionaries, enabling their use for personal, official, and industrial purposes as well as for learning Vietnamese.

This paper has presented an automatic system for diacritic restoration in Vietnamese texts using a pointwise character-based approach. In our experiments, the pointwise approach achieved a high degree of accuracy using three types of features and window sizes of 2 and 3 syllables. We believe that larger window sizes may produce even higher degrees of accuracy. The pointwise approach is simple and language-independent, and it performs well using easily-obtained training data. The accuracy of our approach is not limited to the Vietnamese language; the approach can achieve similar results for any language in which diacritic restoration is a problem.

A negative consequence of building a classifier for each syllable was that the files generated for the model were very large. When the window size was 3 and the training data set size was 320 Mb, the total size of the model's files was 16 Gb. However, we expect that with proper feature selection, the model's files can be made much smaller.

This paper's source code and documentation will be available on the internet on request (http://viet.jnlp.org).

REFERENCES

[1] Graham Neubig, Yosuke Nakata, and Shinsuke Mori, "Pointwise prediction for robust, adaptable Japanese Morphological analysis" In The 49th Annual Meeting of the Associantion for Computational Linguistics : Human Language Technologies, 2011, 529-533.

[2] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification", Journal of Machine Learning Research, 2008, 1871–1874.

[3] Tufiş, D.; Ceauşu, A., "DIAC+: A professional diacritics recovering system" 6th language resources and evaluation conference. ELRA, 2007, 167–174.

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

[5] Simard, M.: "Automatic Insertion of Accents in French Texts", Proceedings of the Third Conference on Empirical Methods in Natural Language Processing, 1988, 27-35.

[6] 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

[7] Nikola Šantić, Jan Šnajder, Bojana Dalbelo Bašić, "Automatic Diacritics Restoration in Croatian Texts", 2nd International Conference, 2007, 309-318.

[8] Mihalcea, R. (2002). "Diacritics Restoration: Learning from Letters versus Learning from Words". In Proceedings of CICLing, 339-348.

[9] Tufiş, D., Chiţu, A. (1999). "Automatic Insertion of Diacritics in Romanian Texts". Proceedings of the 5th International Workshop on Computational Lexicography COMPLEX, Pecs, Ungaria, 1999, 185-194

[10] Phuong, L.H; Huyen, N.T.M; Roussanaly A.; Vinh, H.T; "A hybrid Approach to Word Segmentation of Vietnamese Texts", 2nd International Conference LATA, 2008, 240-249.