SOME ISSUES IN AUTOMATIC SPELL CHECKING OF VIETNAMESE WRITTEN SYLLABLES WITHOUT AN ASSOCIATED SPELLING DICTIONARY

Ngô Thanh Nhàn
March 27, 1994

to be presented at
TUẦN LỄ TIN HỌC 4
the INFORMATICS WEEK 1994 (IW'94)
The Fourth Biennual Technical Conference and Exhibition
Ho Chi Minh City, Viet Nam
August 2-6, 1994

A. INTRODUCTION

Spell checking in Vietnamese is no longer an issue once Vietnamese characters have been finally encoded for use in computers [1,2], and even more so when the knowledge of language is no longer alien to computer scientists. This paper sets out to show that automatic spell checking for the Vietnamese written syllable (*chữ*) is feasible, and goes one step further to show that automatic spell checking without an associated spelling dictionary is also feasible, if not conceptually more consistent and coherent. Furthermore, an automatic spell checking for a set of specialized dictionaries is also desirable, when data entry is mixed (for example, media-wise) and broad-coveraged. That is, one can now think of dictionary compilation with automatic spell checking.

We will show that an independently built dictionary for machine spell checking is conceptually inconsistent, because a spelling dictionary cannot check itself. We will go on to show that a machine spell check can only be at most a sophisticated string searching program against a list (called a dictionary). Such algorithm can only tell whether a word is in the spelling dictionary, but cannot tell whether a word is in the language, or whether it is possible in that language. In order to check the spelling of a spelling dictionary, one must have some knowledge of what constitutes a word in that language.

B. SCOPE OF AUTOMATIC SPELL CHECKING

First and foremost, we ask several questions in order to clarify the subject matter of machine spell checking:

(i) What is spell checking ?
What is spell checking in a particular language ?
What is spell checking in Vietnamese ?

Spell checking, in the context of this paper, is a set of rules which match strings from a natural language text against a list of strings (called a dictionary) the spelling of which is considered "correct". We are thus mainly dealing with computer algorithms.

B.1. *Multi-lingual spell checking*

It is possible to imagine some form of multi-lingual spell checking in the near future. Given a text in electronic form written in any existing natural language, or languages, it is possible to perform spell checking. However, in the present state of computer technology one cannot yet imagine spell checking for a text in both Vietnamese and Russian simultaneously -- for many reasons, including possible overlapping of character codes in an 8-bit environment. Thus, spell checking within the scope of our discussion is spell checking of a text in one natural language. Most of the current spelling softwares are of this type.

B.2. *Spell checking and spelling dictionary*

Spelling software in the market today is essentially a sophisticated and enhanced dictionary lookup. The program looks at one word, or one string, and searches for a match in an associated spelling dictionary for the word or, at best, some limited alternate forms of the word (inflection, gender, number, etc.) and/or limited possible errors.

Spelling based on words is thus limited by the size of the spelling dictionary and the degree of sophistication of the lookup patterns (all possible alternate forms of the searched word, i.e. some form of anagram or morphological shapes). The size of the spelling dictionary is limited for several reasons: (a) new words are coined every day, (b) words in one professional field (or sublanguage) may not be used in any other, and (c) the size of the computer in use. Thus, we can safely assume that for practical reasons, the factory supplied spelling dictionary does not cover all words in a language. Putting all practical limitations aside, for investigation purpose, we are concerned with the theoretical limitation in (a) only.

One additional feature of spell checking is that it allows users to add more words to the spelling dictionary, to overcome factory limitations. Theoretically, a spelling dictionary cannot check the spelling for itself. Thus, words added into the spelling dictionary are assumed to be correctly spelled. Consequently, words in the spelling dictionary are not error-proof, unless there is an independent standard which ascertains the spelling of the spelling dictionary. The independent spelling standard, in this sense, is the main issue of this paper.

Furthermore, adding new words into a spelling dictionary is practically limited to manual entry because it cannot be done automatically unless manual verification of spelling is done.

B.3. ***Spell checking and syntax***

One conceptual limit of spell checking is that a spelling program can only mark off a word which is not in its spelling dictionary. It cannot tell whether the word (in its spelling dictionary) is correct, that is, correctly placed in the sentence. This is of course a syntactic problem, which is beyond the scope of spell checking.

For example, spell checking cannot mark "*sai*" to "*nai*" in "*Tôi bắt được một con sai*". Both "*sai*" and "*nai*" are in standard Vietnamese dictionaries.

B.4. ***Spell checking and automatic correction***

Because of the conceptual limitation of the spelling dictionary, the spelling program can only warn of possible spelling problems, but cannot automatically correct the target word. Several reasons can be surmised: (a) the spelling dictionary is incomplete (the correct word may not be in the dictionary), (b) there may be foreign words intermingled in the text (no correction is possible), and (c) there is more than one possible correction.

Therefore, spell checking only marks the target word for possible spelling errors (i.e. correct patterns may not be in the spelling dictionary) and suggests possible corrections according to existing patterns in the spelling dictionary). Human intervention is needed for such choice.

B.5. ***Possible words and spell checking***

We have argued that in spelling, no associated spelling dictionary is complete, that it is impossible to correct spelling for words newly added, that there is no means to correct spelling for the spelling dictionary, that there is no spell checking of combined morphemes which exist implicitly throughout the spelling dictionary (for example, "*anti-compositionism*" cannot be found in any spelling dictionary, but both "*antizionism*" and "*composition*" may exist).

For the above reasons, we can say that spell checking is a procedure which tells whether a string surrounded by delimiters is in the system (here, the associated spelling dictionary) or not. It does not tell whether such a string is in a (natural) language.

Since Vietnamese has a limited number of written syllables (*chữ*) which are definable, we can define spell checking in Vietnamese as a procedure which tell whether a string (with or without delimiters) is or is not Vietnamese. This is also true for other monosyllabic languages.

B.6. ***Spell checking in Vietnamese***

The nature of the word (*từ*) in Vietnamese is obscure [3]. This fact has been noted by many linguists since Vietnamese was first studied. However, linguistically, the syllable is definable and always holds a special status in Vietnamese (which was called vaguely

as a monosyllabic language). Spell checking in Vietnamese has less to do with words than with syllables. It was noted in [3] that

(ii) A word in Vietnamese, if it exists, consists of one or more syllables (an integer number of syllables).

(iii) A morpheme in Vietnamese, if it exists, consists of one or more syllables (an integer number of syllables).

(iv) Each Vietnamese syllable is written separately and bordered by delimiters. This is usually thought of as a property of monosyllabicity.

(v) A Vietnamese phonological shape corresponds one-to-one with its written shape (cf. Appendix A in [3]).

Since there is a one-to-one correspondence between a phonological [psychologically abstract] syllable (*tiếng*) and a written one (*chữ*), and possible syllables are derived from phonological generation in Vietnamese, it is therefore preferrable to assume that

(vi) Spell checking in Vietnamese is a process of marking non-Vietnamese syllables.

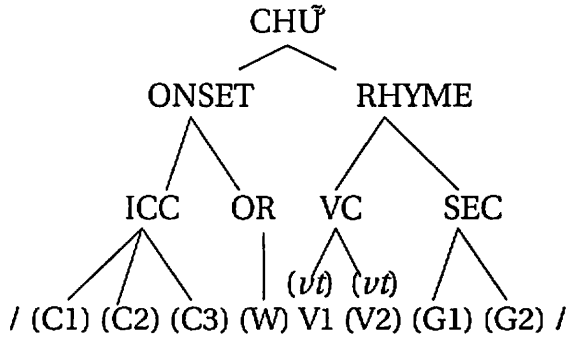
We will prove below that under the assumption (vi), it is possible to devise an algorithm which checks spelling without an associated spelling dictionary.

C. BASIC ASSUMPTIONS AND OBSERVATIONS

A diacritic mark is a tone mark (henceforth, *t*) or vowel mark (henceforth, *v*).

1. Suppose we have a generalized structure of a Vietnamese orthographic syllable [5]:

(1)



One can also write, in a version of BNF formalism, informally:

<CHỮ> ::= [<ONSET>] <RHYME> . (1.1)
<ONSET> ::= <ICC> | <ICC> <OR> | <OR> . (1.2)

<ICC> ::= [<*C1> [<*C2> [<*C3>]]] . (1.3)
<OR> ::= <*W> | <*NULL> . (1.4)
<RHYME> ::= <VC> [<SEC>] . (1.5)
<VC> ::= <*V1> [<**v*>] [<*t*>] [<*V2> [<*v*>] [<*t*>]] . (1.6)
<SEC> ::= <*G1> [<*G2>] . (1.7)

where: <X> non-atomic segmental (non-diacritic) X,
<*X> atomic segmental X;
[X] X or <*NULL>;
| or;
<*v> and <*t>, non-segmental, associated with the left segment.

where: / a syllable delimiter
C1 initial consonant { *b c d đ (f) g h (j) k l m n p q r s t v (w) x (z)* }
C2 supporting initial consonant, { *h g i r* }
C3 supporting initial consonant, { *h* }
W semi-vowel, { *o u* }
V1 central vowel, { *a e i o u y* } and { ă ô ơ }
V2 glide of central vowel, { *a e o* } and { ô ơ }
G1 syllable-ending character, { *i y o u c p t m n* }
G2 supporting syllable-ending character, { *h g* }
t tone mark, { *huyền* ò *hỏi* ỏ *ngã* õ *sắc* ó *nặng* ọ }
v vowel mark, { ă ô ơ }
positions in parentheses are optional.

There are systematic constraints on co-occurrence of elements of a Vietnamese syllable, some of which are relevant to correct diacritic placement [6], and some of which are sufficient for our purpose of syllable checking.

Alternatively, we can also say informally:

(2) Each syllable consists of an obligatory central vowel group, flanked on both sides optionally by consonants.

(3) Each Vietnamese syllable has only either one vowel mark or one single tone mark or both.

In cases such as *cồồng, cưỡng*, there are 2 identical vowel marks in a syllable.

(4) Vowel marks are selective: [ă] only goes with { *a* }, [ô] with { *a e o* } and [ơ] with { *o u* }.

(5) Vowel marks only appear on V1 and/or V2 position, never on G1 position.

Hence, in the Vietnamese writing system, if a syllable has a tone mark and a vowel mark simultaneously, they have to appear on a single vowel position. There is no case where

the vowel mark appears on one vowel while the tone mark appears on another within a single syllable.

(6) Possible errors in vowel and tone placement are in C2 { *i*}, W, V1, V2 and G1 { *i y o u*}.

and more generally,

(7) Tone and vowel diacritics only associate with the last vowel or next to last vowel of a syllable [6].

2. Suppose we have a system of writing Vietnamese in the computer which fuses vowel mark *v*, tone mark *t*, and vowel V into one single ASCII position (i.e. a V*vt* composite, most current Vietnamese systems on PCs are of this sort).

Rules (3) - (5) that are concerned with vowel marks are not relevant under TCVN 5712:1993, and the leaves of rule (1) is simplified for:

(1.a) W semi-vowel, { *o u*}
V1 central vowel, { *a ă â e ê i o ô ơ u ư y*}
V2 glide of central vowel, { *a â e ê o ô ơ*}
G1 syllable-ending character, { *i y o u c p t m n*}

D. SPELL CHECKING IN VIETNAMESE

We say that a Vietnamese syllable is well-formed if and only if a string meets the requirements of the structure in (1) and (1.a) and the following preliminary set of conditions. We call these conditions ***syllable well-formedness conditions*** which operate from left to right together with a loosely-defined parser:

(8) *Initial consonant formation*
i. if C2=*h*, C1={ *c g k n p t*}
This rule gives consonant clusters *ch-, gh-, kh-, nh-, ph-, th-*
ii. if C2=*g*, C1={ *n*}
This rule gives the consonant cluster *ng-*
iii. if C2=*i*, C1={ *g*}
This rule gives the consonant cluster *gi-*
iv. if C2=*r*, C1={ *t*}
This rule gives the consonant cluster *tr-*
v. if C3=*h*, C2={ *g*} and C1={ *n*}
This rule gives the consonant cluster *ngh-*

(9) *Onset rounding dissimilation*
i. if W exists, then ICC does not contain { *b gi m p r v*} and { *c k*}[optional]
ii. if ICC={ *k g*}, then W={ *u*}

The rule (9.i.) is marked optional because there are foreign influenced words such as *Thánh Gioan, moa, cua-roa, voan, puốc-boa, ...* Although we do not have *gu-* for cases like *guy, goai, goăng, guấc, guynh, goe,* etc., they are all possible syllables.

The { *c k* } group is influenced by the Portuguese writing system, where *c, k* and *q* are complementarily distributed, so that *q* is assigned to carry W rounding (i.e. *qu-*).

(10) *Central Vowel rounding dissimilation*
if W exists, then V1 does not contain { *o ô u ư* }.

(11) *Glide and Vowel dissimilation*
i. if V1 = { *a ă e* }, W = { *o* }
ii. if V1 = { *â ê i y ơ* }, W = { *u* }

(12) *VC formation*
i. if V2 exists, V1 = { *a e i o ô u ư y* } [or V1 does not include { *ă â ê ơ* }]
ii. if V2=*a*, V1={ *a i y* u ư }
This rule gives vowel clusters *aa, ia, ya* as in *taáng, hia, khuya, khua, chưa.*
iii. if V2=*e*, V1={ *e* }
This rule gives vowel cluster *ee,* as in *beeng.*
iv. if V2=*ê*, V1={ *i y* }
This rule gives vowel clusters *iê* and *yê,* as in *chiêng* and *chuyên.*
v. if V2=*o*, V1={ *o* }
This rule gives vowel cluster *oo,* as in *boong tàu.*
vi. if V2=*ô*, V1={ *u ô* }
This rule gives vowel clusters *uô* and *ôô,* as in *buông* and *bôồng*
vii. if V2=*ơ*, V1={ *ư* }
This rule gives vowel cluster *ươ* as in *thương.*
viii. if V1={ *ă â* }, V2 = {} and G1 is not {}.
This rule does not allow a syllable to end with *ă* or *â.*

(13) *Portuguese complementary distribution*
i. ICC={ *k gh ngh* }, W={}, VC={ *e ê i y* }
ii. ICC={ *c g ng* }, W={}, VC={ *a ă â o ô ơ u ư* }
iii. if ICC={ *q g* }, W={ *u* }, VC={ *a ă â e ê i ô ơ y* }

(14) *Rhyme dissimilation*
i. if G1={ *i y* }, V1 does not include { *e ê i y* }
ii. if G1={ *o u* }, V1 does not include { *o ô u* }

(15) *Final consonant cluster SEC formation*
i. if G2=*h*, G1={ *c n* }
This rule gives final consonant clusters *-ch* and *-nh.*
ii. if G2=*g*, G1={ *n* }
This rule gives final consonant cluster *-ng.*

(16) *Rhyme formation*
i. if VC={ *ee oo* }, SEC={ *c ng* }
ii. if SEC={ *ch nh* }, VC={ *a ê i y* }
for example, *canh, cách, chênh chếch, huỳnh huỵch...*
iii. if SEC={ *ng* }, VC does not contain { *ê i y ơ* }
however this is a weak rule, since we can imagine *bơng.*

(17) *Occlusive tones*
if SEC={ *c ch p t* }, tone={ *sắc nặng* }

(18) *Orthographic accidents* (perhaps Portuguese complementary distribution)
i. if V1={ *i e ê* }, ICC does not include { *c g ng* }
ii. if V1 or W is one of { *a ă â o ô ơ u ư* }, ICC does not include { *gh k ngh* }

(19) *Generation adjustments to phonological to orthographic equivalences:*
i. *gii-* is adjusted to *gi-*
ii. *quu-* is adjusted to *qu-*

(20) *Minor (dissimilation) adjustments to non-existent rhyme patterns:*
-ău -ăi -ăo -âi -âo -ei -eu -êi -êo -ii -io -iy -yi -yo -yy -ou -ôo -ơo -uu -uo -ưo.
Note that *-io* only exist in *gio.*

The above conditions are incomplete and may overlap in operation. Ideally, the set of rules (1)-(20) would recognize or generate about 15,000 syllables, while in reality, Vietnamese only uses less than 7,000 syllables.

E. EXAMPLES OF SPELL CHECKING

Spelling errors in texts come from many sources, but are not at all at random. Current technology in text input is usually systematic, i.e. depending on the systems of input. For example, the types of errors that handwriting produces is different from those typewriting produces, and these are different from those produced by scanning. Types of spelling errors from electric typewriters are different from those of mechanical (old) ones. American typists have different types of spelling errors than the French, although both may type an English text. We say that both are influenced by the sound and writing patterns of their native language...

E.1. *Anagram and spell checking*

(21) *Assumption 1:*
The word to be checked may have all characters, but not in correct order.

With the help of an anagram, we can suggest the correction of a Vietnamese *chữ* in some case, and give a choice in other cases. For examples, a string of four characters [ê], [i],

[t] and [v] has only one single correct spelling, *viết*, according to (1) and (9)-(18) above.

In this example, we leave out the *sắc* tone, an associated program [6] for correct placement of the tone will do the job.

Thus, an anagram of a string consisting of [v], [i], [ê] and [t] gives only one correct answer, *viêt*:

êitv	êivt	êtiv	êtvi	êvit	êvti
iêtv	iêvt	itêv	itvê	ivêt	ivtê
têiv	têvi	tiêv	tivê	tvêi	tviê
vêit	vêti	*viêt*	vitê	vtêi	vtiê

However, the string of four characters [a], [h], [m] and [n] has two possible correct spellings, *manh* and *nham*:

ahmn	ahnm	amhn	amnh	anhm	anmh
hamn	hanm	hman	hmna	hnam	hnma
mahn	*manh*	mhan	mhna	mnah	mnha
nahm	namh	*nham*	nhma	nmah	nmha

An anagram in the worst case, will create *n!* strings. The worst case in Vietnamese anagrams would be 6! (720 strings), theoretically, much less, in *nghiêng* with many duplicated strings.

E.2. *Anagram, keyboard finger placement and spell checking*

The following assumptions are aimed at arriving at a solution sooner, more for practical reasons than theoretical ones [4]. Thus, for Vietnamese, a system of language specific and keyboard specific assumptions are necessary to direct the choices for spell checking.

(22) *Assumption 2:*
A word to be checked may have all the characters, but not in the correct order and one of the character may be incorrectly entered.

It is possible to show that it is easy to check spelling if the target string (string in question) has all the characters. It is also possible to show that if the target string having more characters than intended, it would be easier to check (correct) spelling than when the string has less characters than intended.

(23) *Assumption 3:*
The word to be checked has all its characters, but not in the correct order and one or more of the characters may be incorrectly entered.

(24) *Assumption 4:*

The word to be checked may not have all its characters entered correctly.

(25) *Assumption 5:*
A typo error usually occurs when fingers are placed in a non-standard manner on the keyboard.

This assumption is useful because with the standard QWERTY keyboard and standard 10-finger typing, an *a* is usually mixed with *q*, *s*, or *z* (left little finger), and an *s* is usually mixed with *a*, *w*, *d* or *x* (left ring finger).

Note that other type of input (such as electronic transfer) may have a different type of errors, sometimes from mixing bits.

E. CONCLUSION

Spelling without a dictionary has important theoretical and practical implications. In Vietnamese, spelling based on the syllable, rather than the word creates strong constraints on the input and thus allows stronger error corrections.

On the other hand, a dictionary is, among other things, a prescriptive standard of spelling. One cannot imagine a dictionary with spelling errors. Massive and mixed data entry in dictionary and database compilation raise a series of other problems pertaining to spelling consistency and correctness, etc. throughout the dictionary/database, which are not at all trivial. Furthermore, any system which requires some form of internal navigation and search for crucial data (such as text databases, encyclopedias, information databases, hypertexts, etc.) also requires a very high level of consistency and correctness in spelling of keys, especially when the keys are actual words in natural language.

REFERENCES

[1] TCVN/Technical Committee. 1993. TCVN 5712:1993 -- The 8-bit Vietnamese Standard Codes for Information Interchange. Hanoi, May 12, 1993.

[2] Đỗ, J, Nhàn, N.T., Hoàng, N. 1992. A proposal for standard Vietnamese character encodings in a unified text processing framework. *Computer Standards & Interfaces* 14:3-10.

[3] Nhàn N.T. 1984. *The syllabemes and patterns of word formation in Vietnamese.* NYU Ph.D. Thesis.

[4] Nhàn, N.T. 1985. Vấn đề kiến trúc chữ Việt trên máy tính điện tử [Some problems in the designing of Vietnamese on a computer], *Đất Việt* 11-1985, also *Vietnam Culture Journal* 4.4: 210-217.

[5] Nhàn N.T. 1986. Một số vấn đề về chuẩn chính tả tự động trong tiếng Việt -- không dùng từ điển [Some issues in the automatic spelling correction in Vietnamese without a dictionary], *Đất Việt* 6-1986: 9-11.

[6] Nhàn N.T., Phước Đ.B., Hoàng N. 1992. Một số kết quả về cách đặt tự động đúng dấu phụ vào chữ viết tiếng Việt [An algorithm for correct placement of diacritic marks on Vietnamese written syllable]. *Tạp chí Ngôn Ngữ* [Linguistic Review], 86:14-23. Hanoi, Vietnam.