Vietnamese Diacritics Restoration as Sequential Tagging

Nguyen Minh Trung
R&D Lab
VNG Corporation
Email: trungnm@vng.com.vn

Nguyen Quoc Nhan
R&D Lab
VNG Corporation
Email: nhannq@vng.com.vn

Nguyen Hong Phuong
School of Information and Communication Technology
Hanoi University of Science and Technology
Email: phuongnh-fit@mail.hut.edu.vn

***Abstract*—Diacritics restoration is the process of restoring original script from diacritic-free script by correct insertion of diacritics. In this paper, this problem is casted as a sequential tagging task where each term is tagged with its own accents. We did careful evaluations on three domains of Vietnamese: writing language, spoken language and literature using two methods: conditional random fields (CRFs) and support vector machines (SVMs), and achieved promising results. We also investigated two levels of lexical: learning from letters and learning from syllables. Although the former performs poorly than the latter, it shows stable results in all three language domains. Therefore, the letter level approach is more useful when we have to deal with unknown words or when words in a sentence are reordered and repeated to achieve stylistic and artistic effect.**

***Index Terms*—Lexical disambiguation, Diacritics restoration, Conditional random fields, Support vector machines.**

I. Introduction

Vietnamese is a language that uses Roman characters in writing together with additional accent and diacritical marks. Normally, we always keep accents attached to syllables while writing Vietnamese texts. However, when people use a computer such as composing emails, chatting, commenting in blogs, they usually skip the accents. The reason is typing diacritics consumes too much time. Most keyboards are designed for English so we need to install Vietnamese keyboard and applies input methods such as Telex, VNI or VIQR to type diacritics. Without accents, the text becomes more difficult to understand for both human and machine. For instance, you build a sentiment analysis system that uses ontology methods to identify and extract subjective information in source materials. How does your system deal with a sentence: [Doi giay dep qua] - a diacritic-free sentence of [Đôi giầy đẹp quá] (These shoes are very nice)? A diacritical ambiguity of the phrase [dep qua] surely causes more difficulties to judge this phrase is positive or negative. Therefore, automatic insertion accents not only increase the results of higher language processing levels but also reduce the typing time when it acts as a diacritical suggestion system.

In this paper, we addressed this problem by considering it as a sequential tagging task where each term is tagged with its own accents. We used two of the most successful statistical learning methods for solving a tagging problem: conditional random fields (CRFs) and support vector machines (SVMs). Experiments were performed on three domains of language: writing language, spoken language and literature, and achieved promising results. We also investigated two levels of lexical: learning from letters and learning from syllables. Although the former performs poorly than the latter, it shows stable results in all three language domains. Therefore, the letter level approach is more useful when we have to deal with unknown words or when words in a sentence are reordered and repeated to achieve stylistic and artistic effect.

The rest of the paper is organized as follows. In Section 2, we present a survey of the previous work in diacritics restoration. Vietnamese diacritics system is briefly described in Section 3. The next part presents the framework of using CRFs and SVMs for restoring accents. The experiments and the result are reported and discussed in Section 5. Finally, we conclude the paper in Section 6.

II. Related Work

Most of the work on accent restoration tackles both the actual task of retrieving diacritics of unmarked text, as well as the related tasks of part-of-speech tagging and word-sense disambiguation. Mihalcea [1] presented a diacritics restoration system that used machine learning methods and operates on the level of the grapheme. It is specially geared towards languages for which no large electronic dictionaries are available. The system is applied to Romanian and achieves accuracy scores of up to 99%. In another approach, Simard [2] using dictionary in combination with learning from words, and accuracy over 90 % was achieved. Nelken and Shieber [3] proposed a weighted finite machine based algorithm. This method employs characters and larger morphological units in addition to words and shows competitive results in terms of accuracy when compared to other methods. Some classification methods include Memory-Based Learning [4], Hidden Markov Models [5]. In Vietnamese, to our knowledge, there are some researches [6], [7], [8] and some software packages for restoring accents, such as AMPad[1] and VietPad[2].

Truyen et al [6] investigated the application of Powered Product of N-grams, Structured Perception and Conditional Random Fields to predicting accents. Using Powered Product of N-grams, they combined n-grams models to estimate the

[1]http://www.echip.com.vn/echiproot/weblh/qcbg/duynghi/ampad
[2]http://vietunicode.sourceforge.net/download/vietpad

language model for correct sequence **v** given the distorted sequence **s**. They also used CRFs for directly modeling $P(v|s)$, the n-grams were used as features instead of the accentless input **s**. Their second-order CRFs achieve best results with about 94.3 % term accuracy on online news domains.

An interesting approach was proposed by Nghia et al [7], in which, they combined both language model and co-occurrence graph to capture the context information within non-accent texts, and then they applied dynamic programming to seek for the original texts that best satisfies the collected context information. For more details, they used the local information (the context within a sentence) to reconstruct an original sentence which is soundness and they also integrated the global information (the content within a whole text - the domain) to avoid ambiguity in semantic meanings. The co-occurrence graph was constructed by applying Apriori algorithm to extract all the common words in a domain and assigning weights according to some support thresholds (minsup). However, the authors did not deal with the task of accent restoration in general but just in a pre-defined domain. They experimented in the domain of informatics.

In the latest experiment, Kiem and Cheol [8] experimented diacritics restoration in Vietnamese using five strategies: learning from letters, learning from semi-syllables, learning from syllables, learning from words, learning from bi-grams. They used C4.5 as classifier and combined AdaBoost and C4.5 to get higher accuracy. Their experiment was performed on education category of VnExpress corpus and achieves 94.7 % in case of using letter-based feature set.

III. Vietnamese Diacritics System

Vietnamese writing system uses a set of alphabetic scripts, which consists of 29 letters (11 vowels and 18 consonants). Among them, 22 letters are latin letters ('f', 'j', 'w' and 'z' letters are eliminated), while the others are new letters. The new letters are created by combining four diacritics with the Roman alphabets (breve, inverted breve, horn, d with stroke). Five tonal marks (acute, grave, hook, tilde, dot-below) are used to represent the tone of a word. There are about 22,000 unique syllables (unigrams) based on this combination. However, in Vietnamese dictionary, only 7,000 syllables are used in writing system to create 40,000 words. About 75% of these words are composed of two syllables, while 7% of these words are composed of more than two syllables.

Unlike other Latin-based languages, Vietnamese has some similar characteristics with the pictographic languages. Specifically, Vietnamese text has no explicit word boundary and the syllables are separated by blank. Therefore, one word may include more than one syllable. For example, the phrase [Bài toán thêm dấu] (Diacritics restoration problem) includes three words: [bài toán], [thêm] and [dấu], while the word [bài toán] has two syllables: [bài] and [toán]. As a result, Vietnamese word segmentation is an important preprocessing phase to perform deeper analysis.

The varieties of Vietnamese letter system where new letters are introduced is one of the most difficult challenges in the Vietnamese diacritics restoring problem. Particularly, the accentless word [doi] may correspond with 26 valid Vietnamese syllables (e.g. [dọi], [đói], [đời], [đồi], ...). To achieve a high accuracy in this problem, we must resolve the diacritical ambiguity (e.g. [o] and [ô], [ơ]) and tonic accents ambiguity (e.g. [o] and [ò], [ó], [ọ], [ỏ]) of syllables. Table I shows all ambiguities in Vietnamese diacritics.

TABLE I
AMBIGUITY IN VIETNAMESE DIACRITICS.

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

IV. Sequential Tagging using CRFs and SVMs

As mentioned earlier, diacritics restoration problem is considered as a sequential tagging task where each word or letter is tagged with its accents. In the next step, we applied conditional random fields and support vector machine to learn and predict the accents.

A. Conditional Random Fields

In this problem, CRFs are referred to as an undirected linear-chain of model states, i.e., conditionally trained finite state machines (FSMs) that obey the first-order Markov property. Let $o = (o_1, o_2, ..., o_T)$ be some observed data sequence. Let S be a set of FSM states, each of which is associated with a label $l \in L$. Let $s = (s_1, s_2, ..., s_T)$ be some state sequence, CRFs [9] define the conditional probability of a state sequence given an observation sequence as:

$$p_\theta(s|o) = \frac{1}{Z(o)} \mathbf{exp}\Big[\sum_{i=0}^{T}\sum_{k}\lambda_k f_k(s_{t-1}, s_t, o, t)\Big] \quad (1)$$

where $Z(o) = \sum_{s'} \mathbf{exp}\big[\sum_{i=0}^{T}\sum_{k}\lambda_k f_k(s'_{t-1}, s'_t, o, t)\big]$ is normalization summing over all label sequences. f_k denotes a feature function in the language of maximum entropy modeling and λ_k is a learned weight associated with feature f_k. Each f_k is either a per-state or a transition feature:

$$f_k^{per-state}(s_t, o, t) = \delta(s_t, l)x_k(o, t) \quad (2)$$

$$f_k^{transition}(s_{t-1}, s_t, t) = \delta(s_{t-1}, l)\delta(s_t, l) \quad (3)$$

where δ denotes the Kronecker-δ . A per-state feature combines the label l of current state s_t and a context predicate, i.e., the binary function $x_k(o, t)$ that captures a particular property of the observation sequence o at time position t. A transition feature presents sequential dependencies by combining the label l'of the previous state s_{t-1} and the label l of the current state s_t.

Training CRFs is commonly performed by maximizing the likelihood function with respect to the training data using advanced convex optimization techniques like L-BFGS. Decoding in CRFs can be done using Viterbi algorithm.

B. Support Vector Machines

Suppose that we have a set of training samples $Tr = \{(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)\}(x_i \in R_n, y_i \in \{-1, +1\})$ where x_i is a feature vector of the i-th sample represented by an n dimensional vector, y_i is the class (positive (+1) or negative (-1)) label of the i-th sample and k is the number of training samples.

In the basic SVMs framework, we try to separate the positive and negative samples by a hyper plane expressed as: $(w \cdot x) + b = 0$. SVMs [10] find and "optimal" hyperplane (an optimal parameter set of w, b) which separates the training data into two classes. In other words, SVM try to the separates hyperplane which maximizes its margin. This problem can be expressed as:

$$\begin{cases} \textbf{Maximize:} & M = \frac{2}{||w||} \\ \textbf{Subject to:} & y_i[(w \cdot x_i) + b] \geq 1 \end{cases} \quad (4)$$

However, the data is not always linearly separable. In that case, a nonlinear mapping function Φ that maps the input data into a higher dimension space is used. In this space, the data can be linear separated, or may be separated with fewer errors than the case using the original space. To avoid calculating directly the integrated scalar $\Phi(x_i) \cdot \Phi(x_j)$, the Kernel Functions $K(x_i, x_j)$ such as Linear, Polynomial, Radian Basis Function and Sigmoid are used.

The reason for choosing SVM to solve this problem is that it was also experimented to identify English base phrases in [11] and showed promising results. This approach used all the information available in the surrounding context, such as the words, their part-of-speech tags as well as the chunk labels. Since the preceding chunk labels are not given in the test data, they are decided dynamically during the tagging of chunk labels. The technique can be regarded as a sort of Dynamic Programming matching, in which the best answer is searched by maximizing the total certainty score for the combination of tags. Then they used pairwise method for multi-class classifier and final decision is given by their weight voting.

V. Experiments

A. Corpus Building

Different from other natural language processing tasks, where annotation of training data and test data is the most time consuming time, the datasets for diacritics restoration are easily collected from the Internet. Therefore, we aim to create a data set that covers almost of the Vietnamese tokens for training and testing phrases. We use a crawler to collect documents from many sources, including informal language, formal language and Vietnamese literature. These documents are selected randomly to increase the variety of our corpus. Furthermore, in each level of lexical, we used a different dataset. There are two reasons for this. First, in syllable level we tagged on each syllable but in letter level we tagged in each letter. Normally, one syllable contains 3-4 letters so that if we use the same number of documents, the tagged dataset for letters is bigger than syllables. Second, the number of letters (29) is much lower than the number of syllables (7,000) so we only need a small corpus to cover all contexts.

1) Writing language: Baomoi[3] is a system that automatically collects articles from about 100 Vietnamese online newspapers: e.g. vietnamnet.vn, vnexpress.net, giaoduc.net, dantri.com.vn, 24h.com.vn, chinhphu.vn, and topics are various, including: World, social issues, culture, economics, science, sport, entertainment, politics, education, health, estate. We crawl about 50,000 articles in this website from September, 2010 through August, 2011 to cover a diverse range of Vietnamese documents. These articles about events, issues in a wide range of time make the data more diversified. In each topic, we randomly get 400 documents from 5,000 documents and have a set of 4,000 articles.

2) Spoken language: Vietnamese spoken language have some typical characteristics. First, sentences are usually short and may lack of subjects. Words are informal and sometimes dialectal (depend on writer's region). Second, language styles often vary with personal interests. Thus, they cause more difficulties for restoring accents.

To gather data for this domain, we chose Tathy[4] - one of the biggest Vietnamese forums is chosen. Tathy was chosen because of two reasons. First, this forum has many posts in informal language. The forum members discuss on the daily events, the cultural issues or politics. Second, due to the forum's law, every post or comment in this forum must be guaranteed that it is written under regular Vietnamese with full accent and no acronym. Finally, after performing some preprocessing, a good corpus of informal language is constructed to test. We retrieved all threads in [Chợ Đuổi][5] - an online auction discussion sub forum for ecommerce, selling, marketing and [Quán nước vỉa hè][6] - an daily news discussion sub-forum.

3) Literature: The most challenging language domain is perhaps creative writing. The authors often use many innovative styles such as short sentences, rhetorical questions or exclamative sentences. Words are frequently reordered and sometimes repeated to achieve stylistic and artistic effect. Some sentences do not obey Vietnamese grammar. In this domain, we focus on short stories, novels and poems.

Vnthuquan website[7] is a good source for literature works. It is an online library with many literature documents, from short stories, long stories, poems to novels. These documents use both informal and formal language. They also have some archaic Vietnamese words. In our experiments, we want to reduce the impact of author style on dataset, so that we did not choose documents based on authors. We randomly retrieved

[3] http://www.baomoi.com/
[4] http://tathy.com/thanglong/
[5] http://tathy.com/thanglong/forumdisplay.php?f=26
[6] http://tathy.com/thanglong/forumdisplay.php?f=28
[7] http://vnthuquan.net/

TABLE II
CORPUS BUILDING FOR EXPERIMENT

Domain Language	Lexical level	No.of Documents	No.of Sentences	No.of Tokens	No.of Unique Tokens	No.of Tokens in Dictionary
Writing language	Syllables	4,000	74,268	2,404,896	24,162	4,340
	Letters	2,000	43,788	1,031,239	14,608	3,923
Spoken language	Syllables	2,000	96,495	1,681,329	21,127	4,200
	Letters	1,000	58,359	1,048,739	19,127	3,530
Literature	Syllables	1,000	47,238	790,250	6,086	3,522
	Letters	800	31,225	746,048	5,231	3,423

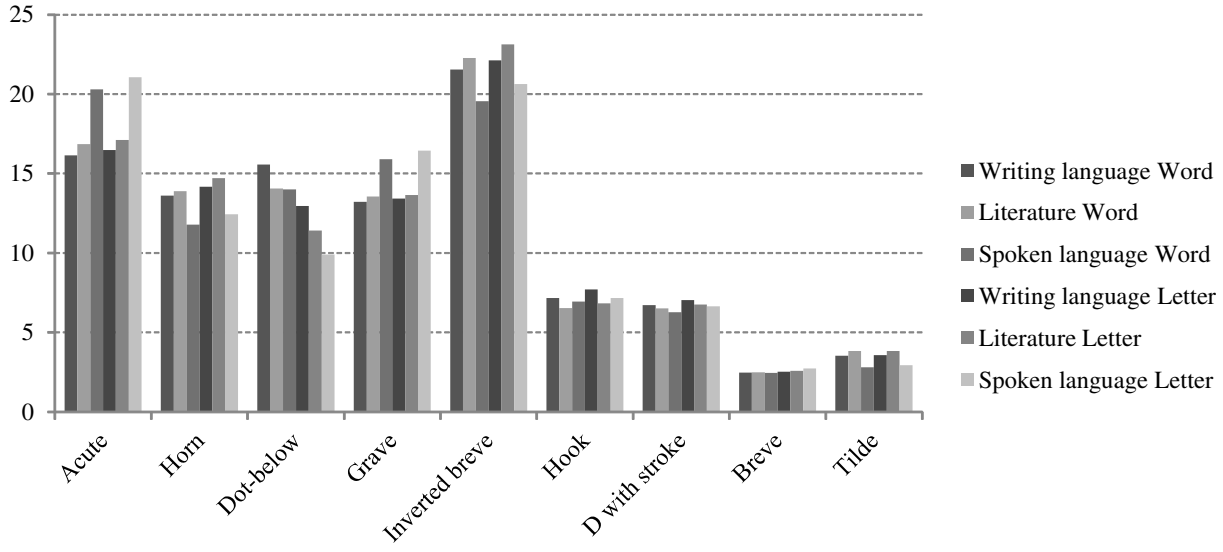

Fig. 1. Diacritics distribution over six datasets

2,000 pages in this website. Each page corresponds to one chapter of a story or one poem.

Finally, after the crawling process, three corpora with 11,963 words in Vietnamese dictionary are generated. In this set of words, 3,061 words are common words. The number of different words is 5,841 words (84 % of Vietnamese unique tokens). This large number of words ensures the variety of our corpus. In each corpus, the number of words which are not included in Vietnamese dictionary is about four to seven times the number of unigram tokens in Vietnamese dictionary. These are foreign words such as English named entities, song titles, country names, locations,... acronym and non-alphabets (number, time, ...) which usually do not contain diacritics. The statistic of three language domains is shown in table II and figure 1.

B. Experimental Setup

To prepare the data for training, we do some preprocessing on the corpus. At first, we only keep characters, which exist in Vietnamese alphabet and seven types of delimiter: space, comma, dot, question mark, semicolon, colon, exclaim mark. These delimiters are tagged as “SP”, “CM”, “DM”, “QM”, “SC”, “CO”, “EM”, respectively. We used dots, question marks and exclaim marks to split the text into sentences. Next, we used some regular expressions to detect regular patterns such as common abbreviations, numbers, dates, times, email addresses, URLs, emoticons. Some patterns were removed such as URLs, email addresses or emoticons since they are useless for assiting insertion accent. Other patterns were tagged with special tags such as Nb (Number), Dt (Date), ...

In syllable level, each syllable in the Vietnamese unigram list was tagged with this accent. It is important to note that, each syllable may contain more than one letter that has accent. In this case, its tag will be a combination of all accents from left to right. In our corpus, 64 combinations are found so we have 64 tags for diacritical syllable. The accentless syllables are tagged with “NA” (no accent). The other unigrams are tagged with “OTH” (other). For instance, the sentence [Tôi đến trường] (I go to school) will be tagged as {Hat, HatAcute, HornHornGrave} .

In letter level, each letter is tagged with it accents. In the same way, 27 tags for letters are used. In our experiments, sentence [Tôi đến trường] will be tagged as {NA, Hat, NA, SP, Stroke, HatAcute, NA, SP, NA, Horn, HornGrave, NA, NA}.

Performance of both the CRFs-based and SVMs-based depends on how well we select features. Table III summarizes the content predict template used in syllable-based and letter-

TABLE III
FEATURE IN CRFS AND SVMS MODEL

No	Syllables	Letters
1	$Syllable$	$Letter$
2	$Syllable_{-2}$	$Letter_{-5}$
3	$Syllable_{-1}$	$Letter_{-4}$
4	$Syllable_{+1}$	$Letter_{-3}$
5	$Syllable_{+2}$	$Letter_{-2}$
6	$Syllable_{-2}Syllable_{-1}$	$Letter_{-1}$
7	$Syllable_{-1}Syllable$	$Letter_{+1}$
8	$SyllableSyllable_{1}$	$Letter_{+2}$
9	$Syllable_{1}Syllable_{2}$	$Letter_{+3}$
10	$IsInitialCapitalization$	$Letter_{+4}$
11	$IsAllCapitalization$	$Letter_{+5}$

TABLE IV
SYLLABLE BASED RESULT USING CRFS AND SVMS

Dataset	CRFs	SVMs
Writing language	0.93814	0.926181
Spoken language	0.910669	0.909621
Literature	0.881903	0.859405

TABLE V
LETTER BASED RESULTS USING CRFS AND SVMS

Dataset	CRFs	SVMs
Writing language	0.910863	0.91685
Spoken language	0.912589	0.91558
Literature	0.899495	0.90331

based experiments. In syllable-based experiment, we used two kinds of features i.e. (i) local context features: syllable-2 .. syllable+2, is the syllable initial capitalization, is the syllable all capitalization (ii) hybrid features: other features in table III. With letter, only local context features were used. Following popular experiment in the literature of lexical disambiguation [1], [8], the window of size 5 to the left and to the right were chosen.

We conducted our experiments using two tools: CRF++[8] a C/C++ implementation of CRFs and Yamcha[9] a C/C++ implementation of SVMs for tagging sequence data. For CRF models, we use first-order Markov dependency, hyper-parameter is 1.5 and cut-off threshold for features is 3. For SVM models, the second degree of polynomial kernel was used. Instead of using pairwise method for multi-class classifier we used one vs all others methods. Although the pairwise classifiers have an advantage of reducing training cost (SVMs requires $O(n^2)$ - $O(n^3)$ training cost, n is the size of training data), they tend to build a large number of binary classifier. In syllable-based level, training (64 * 63) / 2 = 2016 classifiers are required, which clearly cost more time to predict.

For evaluation in each experiment, we use the 5-fold cross-validation test. In particular, we randomly divided the corpus into five partitions. In each fold, one partition is saved for testing while the others function as training data. The final result is the average result of five folds.

C. Experimental Results and Discussion

The result for syllable based approach and letter based approach are shown in table IV and table V, respectively.

$$Accuracy = \frac{\text{Number of term tagged correctly}}{\text{Number of term tagged}}$$

According to table IV, in syllable based experiment, the result of CRFs overcomes the result of SMVs in all three language domains. However, SVMs's accuracy is slightly smaller than CRFs' accuracy, 3% percent in literature and 1% in other domains. Therefore, it shows that SVMs learning is a potential approach to the problem of Vietnamese diacritics restoration. Maybe, with good feature selection, it will give a better performance than CRFs in this task.

It is difficult to compare our results with other results in [6], [7], [8] because the training data and test data are different. Nevertheless, the experimental results (93.81 % in the best case) show that our methods are competitive with other methods for solving this problem.

It can be seen clearly from the figure 2 that the accuracy of the system decreases from writing language to literature, as expected. The more inconsistent word order is, the more difficult the accent prediction is. A significant drop can be observed in literature domain so it strongly confirms our assumption: creative writing is the most challenging domain in three domains.

Discussion about letter level, table V shows that SVMs and CRFs perform nearly equally. In the literature domain, the accuracy of letter-based is higher than syllable-based. However, this fact should not be considered as evidence that the former is more effective than the latter because of two reasons. First, some syllables may contain more than one letter which has diacritics so we need to convert the results to get a reliable comparison. Second, our experiments are conducted in different datasets.

Another interesting conclusion can be drawn when comparing figures 3 to figure 2. In all language domains, the accuracy of letter-based approach is stable and only fluctuates around 91 %. It leads to our belief that we may only need a small dataset to cover all letter's context and this approach is more useful when training data is limited or when we have to deal with unknown word. In case of syllable-based, we need full data to cover all the syllable's context. Therefore, the imbalance of syllable distribution in training data might significantly influence the overall quality of the system.

VI. CONCLUSION AND FUTURE WORK

We have presented an examination of using two sequential tagging methods: conditional random fields and support vector machines for Vietnamese diacritics restoration. Our key contributions of the work is: (i) building a three language domains corpus (writing language, spoken language, literature)

[8]http://crfpp.sourceforge.net/
[9]http://chasen.org/ taku/software/yamcha/

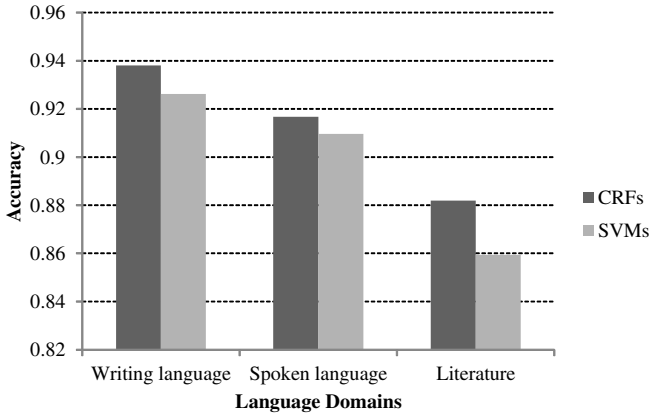

Fig. 2. Syllable based results graph

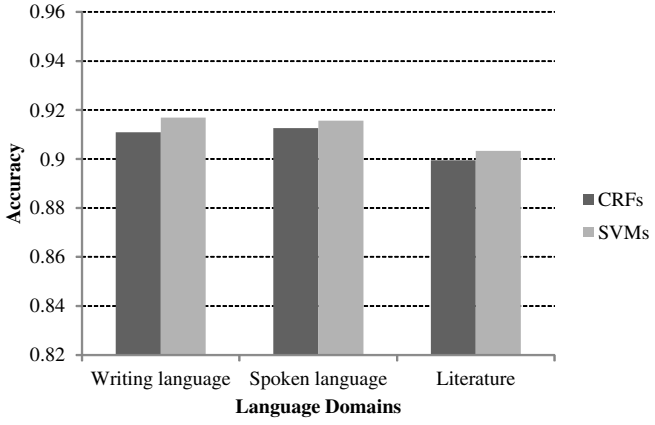

Fig. 3. Letter based results graph

for evaluation; (ii) performing a careful investigation in two levels of lexical: syllable and letter; and (iii) drawing some interesting conclusions from the experimental results.

While our results are encouraging, there are open rooms for further improvement. A global context feature should be extracted from domains and integrating into our framework to increase the final results. We also plan to investigate additional conjunction features and evaluating how well they influence the quality of the system. Finally, an ensemble framework for automatic insertion accent should be built to combine a strong point of each method.

REFERENCES

[1] R. Mihalcea, "Diacritics restoration: Learning from letters versus learning from words," in *Proceedings of the Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002, Mexico City, Mexico*, 2002, pp. 339–348.

[2] M. Simard, "Automatic insertion of accents in french text," in *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing, Granada, Spain*, 1998.

[3] R. Nelken and S. M. Shieber, "Arabic diacritization using weighted finite-state transducers," in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, ser. Semitic '05, Stroudsburg, PA, USA, 2005, pp. 79–86.

[4] G. D. Pauw, P. W. Wagacha, and G.-M. de Schryver, "Automatic diacritic restoration for resource-scarce languages," in *Proceedings of the Text, Speech and Dialogue, 10th International Conference, TSD 2007, Pilsen, Czech Republic*, 2007, pp. 170–179.

[5] M. Simard and A. Deslauriers, "Real-time automatic insertion of accents in french text," *Nat. Lang. Eng.*, vol. 7, pp. 143–165, June 2001.

[6] T. T. Truyen, D. Q. Phung, and S. Venkatesh, "Constrained sequence classification for lexical disambiguation," in *Proceedings of the PRICAI 2008: Trends in Artificial Intelligence, 10th Pacific Rim International Conference on Artificial Intelligence, Hanoi, Vietnam*, 2008, pp. 430–441.

[7] H. T. Nghia and D. Phuc, "A new approach to accent restoration of vietnamese texts using dynamic programming combined with cooccurrence graph," in *Proceedings of the 2009 IEEE, RIVF International Conference on Computing and Communication Technologies*, 2009, pp. 1–4.

[8] K. H. Nguyen and C. Y. Ock, "Diacritics restoration in vietnamese: Letter based vs syllable based model," in *Proceedings of the PRICAI 2010: Trends in Artificial Intelligence, 11th Pacific Rim International Conference on Artificial Intelligence, Daegu, Korea*, 2010, pp. 631–636.

[9] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA*, 2001, pp. 282–289.

[10] V. Vapnik, *The nature of statistical learning theory*. Spinger Verlag, 1995.

[11] T. Kudo and Y. Matsumoto, "Chunking with support vector machines," in *Proceedings of the NAACL 2001*, 2001.