Japanese HWR

Steven B. Poggel
steven.poggel@gmail.com

February 10, 2010

Contents

Summary **5**

Zusammenfassung **7**

1 Introduction **9**
1.1 Motivation . . . 9
1.1.1 Integrating NLP and e-learning . . . 9
1.1.2 Another subsection with a yet unknown title . . . 9
1.2 A CJK environment . . . 9
1.3 Running text . . . 10

2 E-learning **11**
2.1 General E-Learning methods . . . 11
2.2 E-Learning of languages . . . 11
2.3 E-Learning of Japanese . . . 11
2.3.1 Conceptual issues . . . 11
2.3.2 Japanese e-learning software . . . 11

3 Handwriting Recognition Engine **13**
3.1 Error Handling . . . 13
3.1.1 Error Recognition . . . 13
3.1.2 Error Processing . . . 13

4 Implementation and Evaluation **15**
4.1 Implementation Details . . . 15
4.2 Evaluation of the HWR . . . 15
4.2.1 Evaluation Metrics . . . 15
4.2.2 DTW vs. 3-D DTW . . . 15
4.3 Evaluation of E-Learning Application with Integrated HWR . . . 15
4.4 Evaluation of the Error Hints . . . 16

Summary

In this work I present and application that uses state of the art Chinese/Japanese handwriting recognition methods in order to provide an Kanji teaching application with an error correction.

Conceptually, the application is an e-learning environment for Japanese characters, intended for the foreign learner of the Japanese language. In order to provide more than a multiple choice method, like most other systems, the application contains a handwriting recognition engine that can be used preferably with a hand-held device like a PDA, but generally any stylus input device.

Zusammenfassung

In this work I present and application that uses state of the art Chinese/Japanese handwriting recognition methods in order to provide an Kanji teaching application with an error correction.

Conceptually, the application is an e-learning environment for Japanese characters, intended for the foreign learner of the Japanese language. In order to provide more than a multiple choice method, like most other systems, the application contains a handwriting recognition engine that can be used preferably with a handheld device like a PDA, but generally any stylus input device.

Chapter 1

Introduction

1.1 Motivation

In the history of Computational Linguistics there have been a several attempts to integrate natural language processing techniques with existing technologies. This work is one just like that. Concretely, we will try to create a handwriting recognition for Japanese Kanji. That seems interesting, because Kanji is an iconographic writing system, thus handwriting recognition (HWR) can follow different patterns than in alphabetical writing systems like latin.

Studying Japanese language is a complex task, because a new learner has to get used to a new vocabulary that - coming from a European language - has very little in common with the vocabulary of his mother tongue, unlike in European languages where quite often there are several intersections. The learner also needs to learn a new grammar system. Broadly speaking, most of the central European languages follow a subject-verb-object (SVO) structure. Japanese follows a subject-object-verb (SOV) structure therefore creating additional difficulty, comparable with German subclause structures that are a source of error for learners of German. Yet, the most notable difference for a language learner with a central European mother tongue is of course the writing system. The Japanese writing system uses three different scripts. The Kana scripts Hiragana and Katakana are syllabic, each character represents a syllable. Each syllable consists of either a vowel, a consonant and a vowel, or a consonant cluster and a vowel. Hiragana and Katakana represent roughly the same set of syllables and both have around 40-50 characters that can be modified with diacritics and thus yield additional syllable representations. Therefore, these scripts are a hurdle, but relatively unproblematic, due to their limitation in number of characters. Besides, they look quite distinct, so there is the problem of confusing one character with another, but this is limited to a relatively short period of learning those characters.

Kanji, however, is an iconographic writing system that has around 2000 characters, which are built up of around 200 subunits called 'radicals'. So one part of the complexity lies in the number of characters. The other part of the complexity lies in the general concept of representing an idea or concept with a character instead of representing the phonemes of the spoken language with graphemes in connection with some language specific pronunciation rules. Another difficulty lies in connecting the characters with their pronunciations. Most characters have multiple pronunciations and for a language learner, studying Japanese vocabulary is a double or triple task compared to languages using a Latin or at least an alphabetic writing system. Therefore, the two tasks of learning the Kanji and studying the vocabulary together can epitomise a very high learning curve. A subordinated issue connected to that is that quite often subjectively 'simple' vocabulary comes with complex Kanji. Some e-learning applications have taken on that issue by creating a learning environment in which a learner can connect learning vocabulary with studying the Kanji.

xxx: see santosh2009: the statement of need

1.1.1 Integrating NLP and e-learning

In this project, we would like to approach the issue of studying Kanji in an e-learning application. The novelty about it is a handwriting recognition that gives the learner the ability to actually practise writing the Kanji, instead of the rather limited multiple choice recognition that most other applications use.

1.1.2 Another subsection with a yet unknown title

1.2 A CJK environment

Rather than selecting a CJK font as the main document typeface, you might want to define a CJK environment for text fragments used in the midst of a document using a normal Roman font. This allows me to say

`\begin{CJK}`東光`\end{CJK}` to generate 東光, without putting the whole paragraph into the Far Eastern font. Or I could define a command that takes the CJK text as an argument, so that `\cjk{`北京`}` produces 北京. It's that easy!

1.3 Running text

コンピューターは、本質的には数字しか扱うことができません。コンピューターは、文字や記号などのそれぞれに番号を割り振ることによって扱えるようにします。ユニコードが出来るまでは、これらの番号を割り振る仕組みが何百種類も存在しました。どの一つをとっても、十分な文字を含んではいませんでした。例えば、欧州連合一つを見ても、そのすべての言語をカバーするためには、いくつかの異なる符号化の仕組みが必要でした。英語のような一つの言語に限っても、一つだけの符号化の仕組みでは、一般的に使われるすべての文字、句読点、技術的な記号などを扱うには不十分でした。

これらの符号化の仕組みは、相互に矛盾するものでもありました。二つの異なる符号化の仕組みが、二つの異なる文字に同一の番号を付けることもできるし、同じ文字に異なる番号を付けることもできるのです。どのようなコンピューターも（特にサーバーは）多くの異なった符号化の仕組みをサポートする必要があります。たとえデータが異なる符号化の仕組みやプラットフォームを通過しても、いつどこでデータが乱れるか分からない危険を冒すことのなるのです。

Chapter 2

E-learning

- Why this section? The purpose of this section is It would be off purpose, if - What goes into this section? The main content of this section is * if describing a problem: why is the problem relevant. * if describing a solution to a problem: what alternatives were there to solve it, why was this solution chosen? what made it the best choice? was it the optimal solution? - How will this section be structured and organised? The organisational structure of the section - In what style will it be written? The style of writing will be - Next action - what to write first? The next part to write is

2.1 General E-Learning methods

2.2 E-Learning of languages

in section
xxx: s. 12 beachten: WICHTIG!

2.3 E-Learning of Japanese

2.3.1 Conceptual issues

2.3.2 Japanese e-learning software

put all your bashing and criticism here
shortcite nagata2002 not in bibtex yet
(**?**) (**?**) (**?**)

Chapter 3

Handwriting Recognition Engine

3.1 Error Handling

see section **??** in chapter **??** for possible sources of error

3.1.1 Error Recognition

why this section? to demonstrate own achievements of error recognition. the reader should know how it is done technically.

what goes into this section? the aspects of finding errors. finding errors is not a straightforward trivial task - whenever something does not match it is an error - doesn't work like that. instead, firstly, it needs to be made sure that it actually is an error. meaning - not a recognition error, but a user error. secondly, the type of error needs be identified. see section **??** (or handwritten page 58) for sources of error.

how will this section be written? technical - first describe how the error recognition integrates into the recognition process, then how errors are identified.

3.1.2 Error Processing

why this section? actually the 'handling' or 'processing' aspect could be described in the recognition section 3.1.1 as well. so this section is only for a better overview, for document structure, thematically they are the same section. thus they are put together under Error Handling 3.1.

what goes into this section?

Chapter 4

Implementation and Evaluation

- Why this section? The purpose of this section is It would be off purpose, if - What goes into this section? The main content of this section is * if describing a problem: why is the problem relevant. * if describing a solution to a problem: what alternatives were there to solve it, why was this solution chosen? what made it the best choice? was it the optimal solution? - How will this section be structured and organised? The organisational structure of the section - In what style will it be written? The style of writing will be - Next action - what to write first? The next part to write is

4.1 Implementation Details

Pointer auf CD und auf Appendix mit Beispielinteraktionen (diese mit Foto). Screenshots. Zahlen zur Erkennung - z.B. wie lange dauert es, ein zeichen zu erkennen?

wie wurden einzelne dinge realisiert, z.b. vectorielle funktionen? was war neu? klassen wie box / bounding box, technisch, alles was in HWREngine nicht behandelt wurde.

abschnitt ueber optimierung. optimierungszyklus inklusive ausprobieren beschreiben. s. 51 rueckseite

s 49 rueckseite: interface-optimierung entscheidungen herausstellen.

s 27,28 vectorschnitt

s.11 iPhone - port of input app. checked out objective C and stuff!

see section **??**. describe what was difficult concerning the lists and bloody point objects. performance issues! optimisation with try and error!

ISF - see section **??**

dead end of data format description: how I first developed my own format and then found that UPX was better.

in **??** there is an undiscussed point: 3. Description of the production of the lexicon. it was not just taken from j.b. but it was intervowen?! (verflochten) with the trajectories. where did I get these from? how many chars are in the two dictionaries

4.2 Evaluation of the HWR

4.2.1 Evaluation Metrics

evaluation method: counting precision and recall section about precision and recall - the odd numbers. how can that be done honest and useful? how can I get meaningful numbers at all?

s. 12 beachten: WICHTIG: eval kurz und qualitativ.

4.2.2 DTW vs. 3-D DTW

This section al

4.3 Evaluation of E-Learning Application with Integrated HWR

qualitative auswertung, keine zahlen, sondern fragebogen. fuehlt der lerner sich unterstuetzt? glaubt er, dass es schneller geht als ohne HWR? besser als auf papier?

4.4 Evaluation of the Error Hints

use cases, inwieweit helfen die fehlerhinweise? geht das lernen dann wirklich schneller? wie laesst sich der mehrwert bewerten? system kann sagen: wo liegt die verwechslung? warum war das falsch?