CSE4001

Parallel and Distributed Computing

Project Review Report

Name	Registration No.
RSP Varun	*15-BCE-0388*
Rohitanshu Kar	*15-BCE-0550*
Amitya Agarwal	*15-BCE-0568*
Harshit Vats	*15-BCE-0718*
Slot(s): D1	

Faculty: Prof. Manoov R.

School of Computer Science and Engineering

Topic: Parallel Processed Multi-lingual OCR Application

(1) Abstract: There are gigantic endeavours to outline an entire OCR for most of the world's driving dialects, multilingual archives both of manually written or of printed shape. As a unified endeavor, Unicode based OCRs were considered generally with a few positive results, in spite of the way that a huge character set backs off the acknowledgment fundamentally. In this project, we try to mitigate the above defined problem and come out with a strategy to order words to a dialect as the word division is finished. For the reason, the attributes of works of a few dialects were recognized and used anticipating strategy joined with some other element extraction techniques. This project expects a changed factual way to deal with redress the skewness before preparing a portioned report. The proposed system, assessed for an accumulation of both manually written and printed reports, accompanied superb results in appointing words to dialects.

General Terms: Pattern Recognition, Document Processing, Optical Character Recognition.

Classification: OCR, Multilingual OCR, Classification.

(2) Introduction: The main focus of the efforts looking forward for a better recognition of characters in document, always aimed to excel and

achieve a better precision in generating texts from the raw documents. Even an enormous of the researches dedicated to recognize the hand written documents that failed to accomplish 100% accuracy. However, multilingual OCR, a field of prospective is overlooked by and large, in spite of multilingual writing is harnessed worldwide especially in countries where several languages are officially approved.

Ever since the widespread availability of the Penn Treebank, there have been numerous, statistical parsers developed for English, **e.g.** [8, 5, 3]. To varying degrees, these parsers and others—while very successful at the tasks for which they were designed—had the following limitations:

1. They had a fairly fixed probabilistic structure, which could only be changed by re-coding some significant portion of the program
2. They had hard-coded features specific to English
3. They had hard-coded features specific to the Penn Treebank
4. They were designed only for a uniprocessor environment

2.1 The Penn Treebank: The Penn Treebank, in its eight years of operation (1989-1996), produced approximately 7 million words of part-of-speech tagged text, 3 million words of skeletally parsed text, over 2 million words of text parsed for predicate argument structure, and 1.6 million words of transcribed spoken text annotated for speech disfluencies. The material annotated includes such wide ranging genres as IBM computer manuals, nursing notes, Wall Street Journal articles, and transcribed telephone conversations, among others.

No.	Tag	Description
1	CC	Co-ordinating Conjunction
2	CD	Cardinal Number
3	DT	Determiner
4	EX	Existential *there*
5	FW	Foreign Word

Table-1: The above table gives an idea of the alphabetical list of part-of-speech tags used in the Penn Treebank Project.

(3) Background

3.1 Probability structures and sentence-level parallelism

In planning for the work introduced in, we investigated an assortment of lexical likelihood structures including WordNet synsets. To encourage this experimentation, we built up an *attachment 'n'- play* lexical likelihood structure engineering. The model we were expanding was separated from BBN's SIFT framework, which is a history-based model that is gotten in extensive part. For the calculation of likelihood gauges, the BBN parser had hard-coded development of information articles to speak to what's to come what's more, history occasions, figuring a most extreme probability gauge of **P (X|Y)** by **c(X, Y) / c(Y)**, where **c(·)** conveys the include of its contention the preparing information. The engineering of set a layer of reflection at the calculation of the histories for the different back-off levels of the

parser: in pseudo-code, the development of a history question for a specific back-off level turned out to be basically:

```
history = probabilityStructure.get(backOffLevel,
                    fullContext);
```

Efforts have also been made to make an initial foray into parallel-processing by developing a multi-threaded sentence server, to provide parallelism at the sentence level in a cluster computing environment. This work paved the way for the significantly greater degree of parallelism in the architecture of the current parsing engine.

3.2 Language independence

For the project, two parsing models, BBN's SIFT-inferred parser and David Chiang's stochastic TAG parser and adjusted them to parse Chinese. The outcomes were that the Chinese-adjusted models performed with precision near their English partners when prepared on similarly estimated corpora and tried on their particular domains. The principle trouble of the undertaking was uncovering and supplanting all hints of English-particular or Treebank-particular code in the parsers, for example, supplanting code reliant on the default character encoding and finding and abstracting far from language specific word highlights and parts of discourse. This venture extraordinarily affected the dialect free plan of our present design, which takes into account a **dialect bundle** that contains all information and techniques particular to a specific dialect as well as Treebank explanation style.

(4) Design Overview

4.1 Language Package

The most critical information epitome of the plan is that of the dialect bundle. The dialect bundle, which will normally be a real Java package, is a gathering of Java classes that are expansions of a few unique classes which give the specification of information and strategies particular to a specific dialect and Treebank comment style. There are four such required classes: **Treebank**, **Training**, **HeadFinder** and **WordFeatures**. The Treebank class gives all information and techniques particular to a specific Treebank, for example, predicates for pre-terminals and complex non-terminal marks. The Training class gives techniques and information particular to pre-processing preparing trees. The essential capacity of the HeadFinder class is to peruse an information (content) record indicating head rules particular to a specific dialect's Treebank and give a head-discovering strategy. At long last, the WordFeatures class gives a mapping of lexical things of a specific dialect to orthographic/morphological word-feature– vectors, to help grammatical form tagging. As an extra dialect free outline measure, all pertinent info and yield documents are composed to and perused from with a client settable character encoding.

4.2 Probability structure objects

Each kind of yield component of the models bolstered by the designed motor—including non-terminals, preterminals (parts of discourse), words, word-highlights, holes and subset outlines—has a related

ProbabilityStructure question, that indicates how to shape the information objects speaking to the future and history at all conceivable back-off levels for that yield component. Items containing maximal setting—called TrainerEvent objects—are passed in to a solitary strategy which at that point conveys the fitting history or future for a indicated back-off level, according to the pseudo-code case. A layer of reflection likewise exists for these probabilistic occasions: protests that speak to occasions must fit in with an Event interface. This reflection takes into account greatest adaptability in the hidden sorts and portrayal of occasion components.

4.3 Probability-level parallelism

The least complex type of parallelization is to expand a parser's throughput by part up a test record into various fragments and have different executable keep running on these sections, each on a different requirement. The sentence server gave a better granularity of parallelism, yet at the same time required a different parser on each host. Plans for example, these are an amazing misuse of memory assets: each parser contains the same monstrous tables containing tallies of occasions seen in the preparation information.

(5) MOCR Operation Principle

5.1 Projection Discrimination

Category 1: Languages that use English alphabets or modified (alike) form of English alphabets. Languages in this class are listed in Table 2.

English	Latin	Italian	French
Spanish	Danish	Portuguese	German
Croatian	Turkish	Norwegian	Dutch
Polish	Latvian	Hungarian	Dutch
Finnish	Romania	Vietnamese	Czech
Slovenian	Swedish	Indonesian	Irish
Hungarian	Filipino	Slovak	

Table-2: List of languages that use English or alike alphabets.

*Category 2 (**SASB Languages**):* Languages in this category use Stroke based alphabets; most of those belong to South Asia. Languages in this class are listed is Table 3. The phrase **South Asian Stroke Based Languages** is abbreviated to SASB Languages.

Bengali	Hindi	Nepali
Punjabi	Sanskrit	Marathi

Table-3: List of Languages that use Stroke based alphabets

Corollary 1: *The range where leading (upper amplitude) values belong in the profile, is in a distance than the class of values in previous larger range or the overall average amplitudes for stroke based languages. The range, in addition belongs to upper half of the projection for SASB languages.*

Corollary 2: *Intra-word spacing is absent (more or less) or negligible in South Asian Stroke based languages (Category 2).*

5.2 Twofold Classifier

The twofold classifier, if employed, checks both the horizontal and vertical projections to confirm the classification of a word. In case of multilingual printed documents written in languages in category 1 & 2; the method out performs as printed documents always restrain be intra-word spacing however multilingual handwritings as the slant is user dependent and character adjoin may cause the corollary 2 unviable. Therefore, printed documents validates both the corollaries however handwritings in some cases invalidate corollary 1.

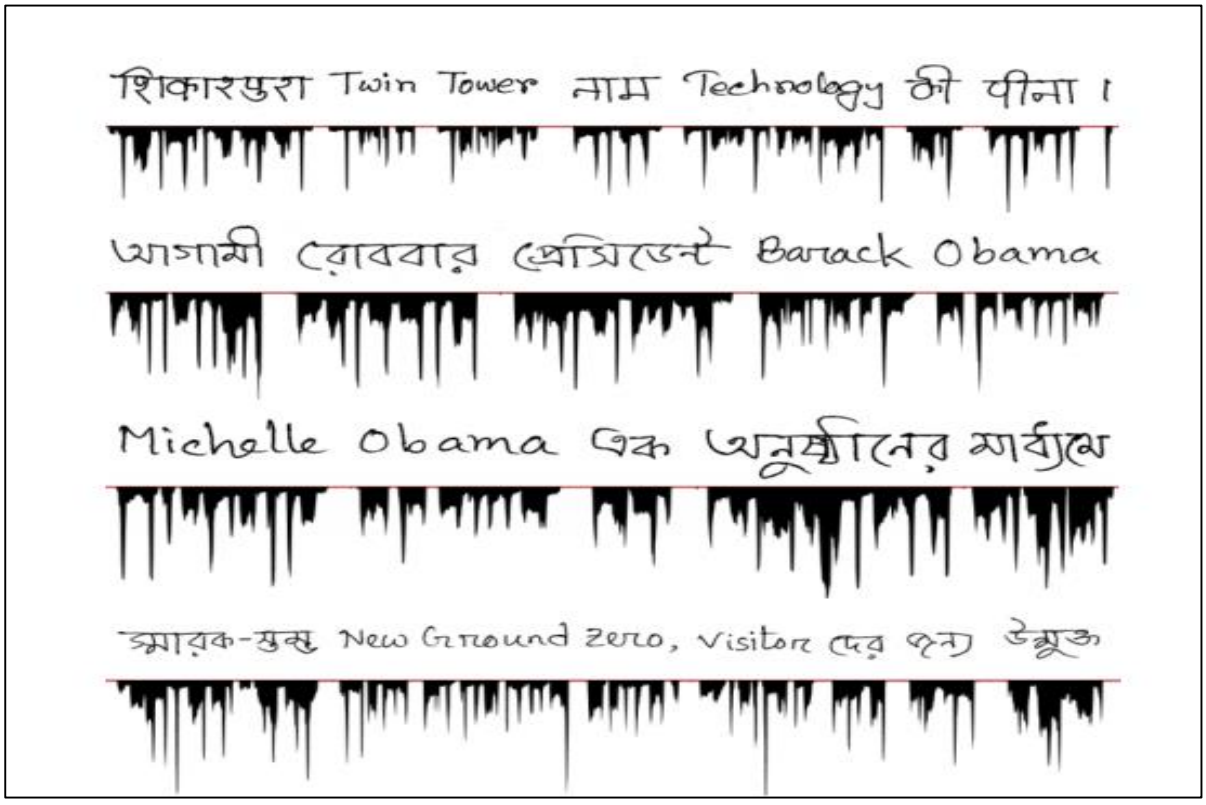

Figure: Vertical Projection for some multilingual text lines.

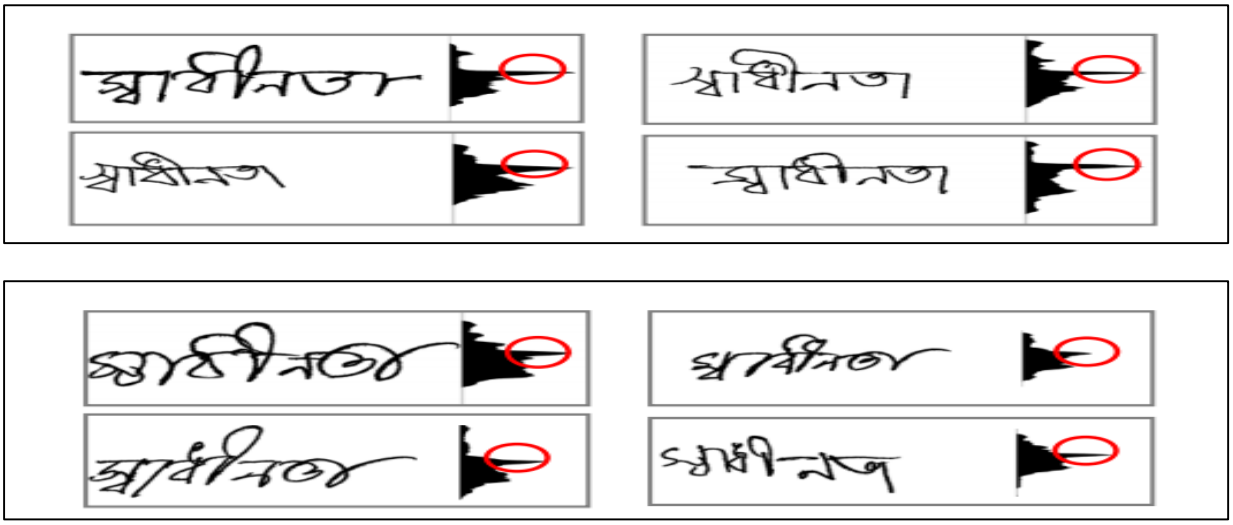

Figure: Horizontal projection of a word written in 8 different handwriting styles.

Segmentation and Classification					
	Docs.	**Line**	**Word**	**Classify**	**Error**
Guided Writings	64	1076/1088 (98.9%)	7513/7424 (1.4%)	SASB: 4729/4800 (98.5%)	6.1%
Unguided Writings	57	923/1054 (97.3%)	6823/6612 (3.2%)	SASB: 3856/4275 (90.2%)	27%
Random Writings	52	678/647 (4.8%)	2854/2707 (5.4%)	SASB: 1912/1970 (97.1%)	27.8%
Total	173	2677/2789	17190/16743	SBSB: 10497/11045	

Table-4: Performance Appraisal of Proposed Classifier.

Snippet of the code file *process_image*

```
#!/usr/bin/python
import cv2
import numpy as np
import sys
import os.path

if len(sys.argv) != 3:
    print "%s input_file output_file" % (sys.argv[0])
    sys.exit()
else:
    input_file = sys.argv[1]
    output_file = sys.argv[2]

if not os.path.isfile(input_file):
    print "No such file '%s'" % input_file
    sys.exit()
DEBUG = 0

# Determine pixel intensity
# Apparently human eyes register colors differently.
# TVs use this formula to determine
# pixel intensity = 0.30R + 0.59G + 0.11B
def ii(xx, yy):
    global img, img_y, img_x
    if yy >= img_y or xx >= img_x:
        #print     "pixel     out     of     bounds
("+str(y)+","+str(x)+")"
        return 0
```

```
    pixel = img[yy][xx]
    return 0.30 * pixel[2] + 0.59 * pixel[1] + 0.11 *
pixel[0]

# A quick test to check whether the contour is
# a connected shape
def connected(contour):
    first = contour[0][0]
    last = contour[len(contour) - 1][0]
    return   abs(first[0]   -   last[0])   <=   1   and
abs(first[1] - last[1]) <= 1

# Helper function to return a given contour
def c(index):
    global contours
    return contours[index]

# Count the number of real children
def count_children(index, h_, contour):
    # No children
    if h_[index][2] < 0:
        return 0
    else:
        #If the first child is a contour we care about
        # then count it, otherwise don't
        if keep(c(h_[index][2])):
            count = 1
        else:
            count = 0
```

(6) Conclusion

Almost all of the techniques and technologies used in this project can be applied to NLP software in general, particularly in light of the growing need to develop efficient, internationalizable software. By abstracting away from the idiosyncrasies of training data and from a particular probability structure for parser output elements, the project can provide parser developers a means to test various probability structures, using features and back-off schemes that are the result either of pure theorizing or empirical study or (presumably) both, and this experimentation is allowed while maintaining a high degree of computational efficiency.

Whether or not morphological features help parsing also depends on the kind of model in which they are embedded, and the different ways they are treated within. Furthermore, sound statistical estimation methods for morphologically rich, complex lexica, turn out to be crucial for obtaining good parsing accuracy when using general-purpose models and algorithms. In the future we hope to gain better understanding of the common pitfalls in, and novel solutions for, parsing morphologically ambiguous input, and to arrive at principled guidelines for selecting the model and features to include when parsing different kinds of languages.

(7) References

[1] Daniel M. Bikel. A statistical model for parsing and word-sense disambiguation. In Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Hong Kong, October 2000.

[2] Daniel M. Bikel and David Chiang. Two statistical parsing models applied to the Chinese Treebank. In Martha Palmer, Mitch Marcus, Aravind Joshi, and Fei Xia, editors, Proceedings of the Second Chinese Language Processing Workshop, pages 1–6, Hong Kong, 2000.

[3] Eugene Charniak. A maximum entropy–inspired parser. In Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics, pages 132–139, Seattle, Washington, April 29 to May 4 2000.

[4] David Chiang. Statistical parsing with an automatically-extracted tree adjoining grammar. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, 2000.

[5] Michael Collins. Three generative, lexicalised models for statistical parsing. In Proceedings of ACL-EACL ’97, pages 16–23, 1997.

[6] Michael John Collins. Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania, 1999.

[7] D. Magerman. Statistical decision tree models for parsing. In 33rd Annual Meeting of the Association for Computational Linguistics, pages 276–283, Cambridge, Massachusetts, 1995. Morgan Kaufmann Publishers.

[8] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics, 19:313–330, 1993.

[9] Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. SIFT – Statistically-derived Information From Text. In Seventh Message Understanding Conference (MUC-7), Washington, D.C., 1998.

[10] Fei Xia, Chung-hye Han, Martha Palmer, and Aravind Joshi. Comparing lexicalized treebank grammars extracted from Chinese, Korean, and English corpora. In Martha Palmer, Mitch Marcus, Aravind Joshi, and Fei Xia, editors, Proceedings of the Second Chinese Language Processing Workshop, pages 52–59, Hong Kong, 2000.

[11] Ryan T. McDonald, Koby Crammer, and Fernando C. N. Pereira. 2005. Online large-margin training of dependency parsers. In Proc. of ACL'05, Ann Arbor, USA.