Impact of imperfect OCR on part-of-speech tagging

Xiaofan Lin
Intelligent Enterprise Technology Laboratory
HP Laboratories Palo Alto
HPL-2002-7 (R.1)
November 18th , 2002*

E-mail: xiaofan.lin@hp.com

part-of-speech tagging, optical character recognition, natural language processing, system combination, majority voting, sensitivity analysis

Part-of-speech (POS) tagging is the foundation of natural language processing (NLP) systems, and thus has been an active area of research for many years. However, one question remains unanswered: How will a POS tagger behave when the input text is not error-free? This issue can be of great importance when the text comes from imperfect sources like Optical Character Recognition (OCR). This paper analyzes the performance of both individual POS taggers and combination systems on imperfect text. Experimental results show that a POS tagger's accuracy will decrease linearly with the character error rate and the slope indicates a tagger's sensitivity to input text errors.

* Internal Accession Date Only
Approved for External Publication
© Copyright Hewlett-Packard Company 2002

Impact of imperfect OCR on part-of-speech tagging

Xiaofan Lin

Hewlett-Packard Laboratories, 1501 Page Mill Road, MS 1126, Palo Alto, CA 94304
Email: xiaofan.lin@hp.com

Abstract

Part-of-speech (POS) tagging is the foundation of natural language processing (NLP) systems, and thus has been an active area of research for many years. However, one question remains unanswered: How will a POS tagger behave when the input text is not error-free? This issue can be of great importance when the text comes from imperfect sources like Optical Character Recognition (OCR). This paper analyzes the performance of both individual POS taggers and combination systems on imperfect text. Experimental results show that a POS tagger's accuracy will decrease linearly with the character error rate and the slope indicates a tagger's sensitivity to input text errors.

1. Introduction

Part-of-speech (POS) tagging tries to assign the proper lexico-grammatical wordclass annotation to each word in the input text based on its lexical meanings and context. It is a very basic problem in the whole chain of the natural language processing (NLP) and has a wide range of applications such as syntactic parsing, noun phrase recognition, keyword extraction, and information extraction. Considering the theoretical and practical value of POS tagging, researchers have developed numerous successful methods for the task: Hidden Markov Models (HMM) [9], transformation based learning [6], memory based learning [8], maximum entropy [7], and so on. Most effort has been concentrated on the intrinsic linguistic aspects: how to resolve the ambiguity when a word can have several syntactic roles, and how to make the best guess when facing an unknown word. It is always assumed that the text itself is free of errors.

However, this "perfect text" assumption cannot hold true in many real-world applications. Figure 1 shows a typical content understanding system, which consists of two subsystems: digitization and NLP. The digitization subsystem re-masters documents not directly available in digital form. For example, Optical Character Recognition (OCR) is employed to convert the image of a printed document back into text in ASCII or UNICODE format. As in any computerized pattern recognition, errors are inevitable even with the smartest algorithms and the most powerful computers. In some applications, a manual proofing step will follow the automatic recognition to guarantee a near-zero error rate. In many other applications, the recognition results will be accepted as-is due to efficiency and/or cost concerns, and thus the text passed into NLP subsystem is imperfect.

The errors from the digitization step will certainly adversely affect the accuracy of the NLP subsystem. The question interesting to us is: What is the extent of influence? This paper addresses this little-studied and yet important problem. In Section 2, the problem is analyzed in detail and a "black box" based evaluation method is proposed. In Section 3, the experiment setup is presented. In Section 4, several observations are made based on the experiments. The last section gives a summary of the paper.

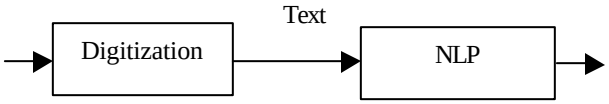

Figure 1: Typical Content Understanding System

2. Problem analysis

- **Error Sources**

Errors in the input text can affect the following POS tagging in a few ways. A word becomes a "strange" word if there are misrecognized character(s) in it. The tagger cannot find the word in the lexicon and is prone to make an error guessing its syntactic role. Worse still, the wrong word sometimes becomes another legitimate dictionary word and "traps" the tagger into a mistake. These tagging errors are called "direct errors" because they happen at the misrecognized words themselves. In addition, because all the POS taggers utilize context information either in probability models or in a series of rules, a wrong word can leads to incorrect tagging of neighboring error-free words. Such errors are "indirect errors."

- **Error Spectrum**

There are two extremes in the whole spectrum. In the worst scenario, when there are so many errors in the input text that even human beings cannot understand the meaning of the text, no computerized taggers can be expected to perform well. In the best scenario, when the text is perfect, researchers have already proposed quite a few efficient solutions (see Section 1). What interests us most is the situation in the middle: How will the taggers perform when faced with different degrees of errors in the input text? Obviously, because taggers employ different mechanisms, they are expected to react to those errors to different extents.

- **Approaches**

There two approaches towards the evaluation of the impact. One is the "white box" approach, in which a mathematical model is built on how input errors affect tagging. This can be a very complicated if ever possible task. The other one is the simpler "black box" approach, in which the tagger is directly tested on input text with various degrees of errors. Although this method cannot tell us exactly why the tagger is affected, it does answer the question we are more concerned about: How much the tagger is affected? So this paper adopts the "black box" methodology.

3. Experiment setup

- **POS Taggers**

Three individual taggers are examined to represent the typical techniques used in POS tagging. The first tagger is the Transformation Based Learning system developed by Brill Eric [6]. It first tags each word with the most likely tag and then modifies the tagging according to learned transformation rules. The second one is the NLP Processor developed by Infogistics [10]. As a commercial product, it does not provide much insight on the technology behind the scene. According to the scattered information on the website, this system seems to follow the traditional trigram model. The last tagger is the QuickTag from Cogilex R&D Inc [11]. The website explicitly claims that "Tagging is not done on a statistical basis but on linguistic data (dictionary, derivational and inflectional suffixes, prefixes, and, derivation rules)."

In addition, to reflect the trend of combining multiple taggers, we have also built a combination tagger based on the above three systems. In recent years, combining several complementary individual systems is a hot topic in NLP and researchers have reported a number of promising results in POS tagging [1][4][5], noun phrase recognition [3], and parsing [2]. In this paper, weighted majority voting, which is similar to "TotPrecision" [1][5], is used due to its simplicity, wide applicability and performance comparable with more sophisticated strategies. In our experiment, Brill tagger is statistically the best among the three taggers, so we assign a weight of 1.1 to Brill tagger and 1 to the other two taggers.

- **Choice of Tagset**

Before proceeding further we have to decide on the tagset for use. A tagset is composed of all the legitimate wordclass tags. After several decades' research, quite a few popular tagsets have been established in the NLP community. They vary in the extent of details and the number of tags. It is common that one tag in Tagset A corresponds to several tags in Tagset B. Because there is no objective standard on which tagset is best, most of the time the tagset is selected on the basis of tradition, testing corpora and convenience. We select the Treebank Tagset developed by University of Pennsylvania [12] since two out of the three taggers (Brill Tagger and NLP Processor) natively support it. We only need to map the proprietary tags in the Cogilex tagger to their correspondence in Treebank Tagset.

- **Corpus**

The testing corpus comes from the AMALGAM Project [13] and includes the text from three sources: the Industrial Parsing of Software Manuals (IPSM), the Lancaster/IBM Spoken English Corpus (SEC), and the Corpus of London Teenager (COLT). There are 180 sentences or more than 3,500 words in the corpus.

- **"Black Box" Testing**

The original corpus is considered as error-free. For the purpose of "black box" testing, errors are artificially introduced into the text. We select a series of character error rates. For each error rate (P), we randomly modify the characters in the original text with a probability of P. Then the tampered text is passed to three individual taggers and their tagging results are combined using weighted voting (Figure 2).

Table 1 is an example when the character error rate is 5%. Two of the five errors in the input text introduce tagging errors and the remaining errors do not affect the tagging. Table 2 shows the statistics. The first column shows the character error rates and the remaining columns display the taggers' error rates.

4. Analysis of experimental results

There are a few interesting characteristics in the testing results:

- Figure 3, the graphic representation of data in Table 1, reveals that the tagger's error rate approximately increases linearly with the character error rate and the slope reflects a tagger's sensitivity to input errors. The slopes of the taggers are shown in Table 3. The greater the slope is, the more sensitive the tagger is to input text errors. So we define the slope as Input Error Sensitivity (IES).
- When the input text is perfect, the weighted voting can reduce the tagging errors significantly (by more than 20% in this experiment). However, with more and more errors in the input text, the improvement is more and more limited. When the character error rate is above 5%, the weighted voting is even worse than the best individual tagger (Brill Tagger). This phenomenon shows the importance of high-quality digitization as an upstream subsystem.
- A tagger's high accuracy on perfect text does not guarantee its resistance against input text errors (Table 3). For example, NLP Processor is the second best individual tagger when the input text is error-free, but its performance deteriorates the fastest when the character error rate increases. When the character error rate is above 8%, its accuracy is the worst among all the taggers. Similarly, the IES of the weighted voting ranks the second largest, although it enjoys the highest accuracy on perfect text.

5. Conclusions

We have proposed the "black box" approach to evaluating the effects of imperfect OCR on the NLP. More specifically, we have examined how the errors in the input text will affect the POS tagging. We have shown experimentally that the quality of OCR directly affects the performance of the NLP in the complete content understanding system. If OCR is too bad, extra effort in the NLP subsystem, including the combination of multiple systems, can be futile.

6. Acknowledgements

The author would like to thank Igor Boyko for obtaining the NLP Processor, Brill Tagger, and the corpora, Steven Simske for valuable comments.

7. References

[1] H. V. Halteren, J. Zavrel, and W. Daelemans, "Improving Data Driven Wordclass Tagging by System Combination", Proc. COLING/ACL, Montreal, Canada, Aug 1999, pp. 491-497.

[2] J. C. Henderson, "Exploiting Diversity", Ph. D Thesis, Johns Hopkins University, Aug 1999.

[3] E. F. Tjong and K. Sang, "Noun Phrase Recognition by System Combination", Proc. of ANLP-NAACL, Morgan Kaufman Publishers, Seattle, Washington, 2000.

[4] E. Brill and J. Wu, "Classifier Combination for Improved Lexical Disambiguation", Proc. of COLING-ACL, 1998.

[5] H. V. Halteren, J. Zavrel, and W. Daelemans, "Improving in Wordclass Tagging through Combination of Machine Learning Systems", Computational Linguistics, 2001, 27 (2), pp 199-230.

[6] E. Brill, "A Simple Rule-based Part-of-speech Tagger", Proc. of the Third ACL Conference on Applied NLP, Trento, Italy, 1992, pp. 152–155.

[7] A. Ratnaparkhi, "A Maximum Entropy Part-of-speech Tagger", Proc. of the Conference on Empirical Methods in Natural Language Processing, University of Pennsylvania, May 17-18, 1996.

[8] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis, "MBT: A Memory-based Part of Speech Tagger Generator", Proc. of Fourth Workshop on Very Large Corpora, 1996, pp. 14–27.

[9] T. Brants, "TnT – A Statistical Part-of-speech Tagger", Proc. of ANLP-NAACL, Morgan Kaufman Publishers, Seattle, Washington, 2000.

[10] http://www.infogistics.com.

[11] http://www.cogilex.com.

[12] Mitchell P. Marcus, Beatrice Santoriniy, and Mary Ann Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank", Computational Linguistics, 1993, vol 19 no 2, pp. 313–330.

[13] http://www.comp.leeds.ac.uk/amalgam/

amalgam/ amalghome.htm.

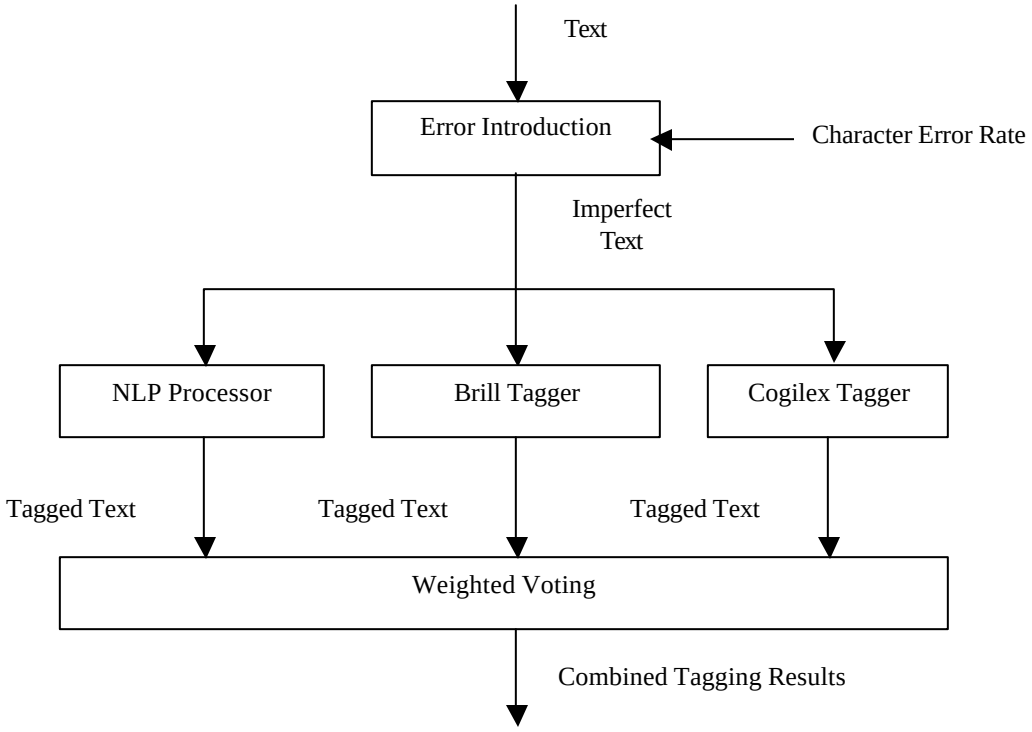

Figure 2: “Black Box” Testing Workflow

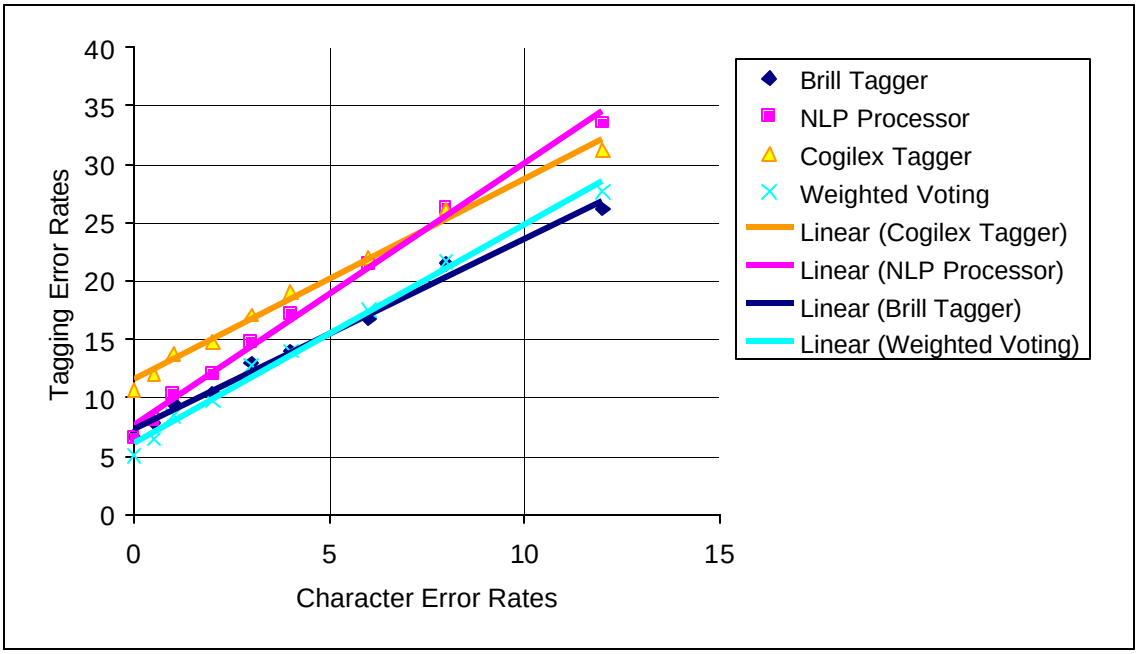

Figure 3: Linear Fitting on the Testing Results

Table 1: Example of Effects Text Errors Have on Tagging Results
(The first row is the original text with correct POS tags. In this example, all the taggers get the correct results on the error-free text. In the second row, the text errors are underlined. In the remaining rows, the incorrectly tagged words are underlined, and the correctly tagged words with incorrect characters are in bold typeface.)

Original Text with Correct POS Tags	Things/NNS do/VBP change/VB in/IN this/DT part/NN of/IN the/DT world/NN in/IN the/DT most/RBS unbelievable/JJ ways/NNS
Text with Errors	Things do chanTe in this pqrt of the world in the mosm unwelCevable ways
Brill Tagger	Things/NNS do/VBP chanTe/NN in/IN this/DT pqrt/**NN** of/IN the/DT world/NN in/IN the/DT mosm/NN unwelCevable/JJ ways/NNS
NLP Processor	Things/NNS do/VBP chanTe/NNP in/IN this/DT pqrt/**NN** of/IN the/DT world/NN in/IN the/DT mosm/JJ unwelCevable/JJ ways/NN
Cogilex Tagger	Things/NNS do/VBP chanTe/NN in/IN this/DT pqrt/**NN** of/IN the/DT world/NN in/IN the/DT mosm/NN unwelCevable/JJ ways/NNS
Weighted Voting	Things/NNS do/VBP chanTe/NN in/IN this/DT pqrt/**NN** of/IN the/DT world/NN in/IN the/DT mosm/NN unwelCevable/**JJ** ways/NNS

Table 2: "Black Box" Testing Results

Character Error Rates (%)	Brill Tagger Error Rates (%)	NLP Processor Error Rates (%)	Cogilex Tagger Error Rates (%)	Weighted Voting Error Rates (%)
0.0	6.601391	6.648939	10.65501	5.04502
0.5	7.892614	8.069415	11.97789	6.430418
1.0	9.515195	10.31622	13.7682	8.46303
2.0	10.40556	12.00926	14.8037	9.715414
3.0	12.95487	14.74638	17.10893	12.72316
4.0	13.99841	17.21688	19.06806	14.01645
6.0	16.79662	21.45045	21.94593	17.4538
8.0	21.48569	26.25987	26.16537	21.66885
12.0	26.15713	33.52958	31.24594	27.64517

Table 3: Input Error Sensitivity Vs. Tagging Error Rate on Perfect Text
(The numbers in the parentheses are the ranking orders)

Taggers	Brill Tagger	NLP Processor	Cogilex Tagger	Weighted Voting
Input Error Sensitivity	1.6263 (1)	2.2458 (4)	1.7168 (2)	1.8671 (3)
Error Rate on Perfect Text	6.601391 (2)	6.648939 (3)	10.65501 (4)	5.04502 (1)