Identifying Part-of-Speech Patterns for Automatic Tagging

Lynellen D. S. Perry, Dept. of Computer Science, Mississippi State University, MS, USA

Abstract

Some part-of-speech tagging errors are very damaging to the ability to further process the text. For systems that use part-of-speech tagging as a prelude to parsing and knowledge extraction, it is imperative to have the cleanest possible tagging. A state-of-the-art rule-based tagger has an error rate of approximately 39% when annotating main verbs that have not been previously seen. We apply neural networks to this real-world problem of identifying part-of-speech patterns that indicate a main verb so as to correct the output of the rule-based tagger.

Introduction

[3] and [4] more thoroughly detail the problem under inspection in this paper. Briefly, we have developed a system which automatically generates text indices for journal articles in the field of physical chemistry [2, 8]. The front end of this system is a module that receives raw text and outputs processed text. The processing includes breaking the text into distinct tokens, annotating each token with a label (part-of-speech or other label as appropriate), and creating a shallow parse tree for each sentence. The parse trees are then handed to a module which carries out a knowledge extraction algorithm and generates the indices for the journal article.

The state-of-the-art rule-based Brill tagger [5, 6, 7] makes mistakes as it goes along, some of which are irrelevant to further processing and some of which are devastating. One very damaging error is to mis-label the main verb in a sentence. When the main verb is mis-labeled, the entire sentence is typically lost and is unavailable for use in knowledge extraction. In our testing corpus, approximately 39% of the previously unseen main verbs were incorrectly labeled by a state-of-the-art part-of-speech tagger. We attempt to correct most of these main-verb errors by applying neural networks to recognize complex part-of-speech context patterns.

Experimental approach

Previous research has shown that neural networks can be effective at recognizing complex part-of-speech patterns and assigning the correct part-of-speech to words in the text [1, 3, 4]. Presented here is a comparison of several neural network architectures we have examined for use on this problem. For a simpler but similar problem, [1] found that they needed a fully connected neural network with nearly 300 hidden units in order to have adequate performance. Since we tackle a larger problem space, we decided to compare the performance of networks with 300 hidden units to the performance of even larger networks with 500 hidden units.

In the first set of experiments, each network was fully connected. The architecture difference between networks consisted of how many input nodes existed. The input nodes represent the part-of-speech context around the word to be tagged by the network. Brill's tagger examines a context of up to three words on each side of the target word but many part-of-speech patterns are longer than this [3]. Accordingly, we examined a variety of context sizes. In all the tables below, a code of L2 R0 means that there were two part-of-speech tags on the left of the target word, and no part-of-speech tags in the right-hand context. Similarly, the code L6 R4 means that six part-of-speech tags made up the left context while four part-of-speech tags were in the right-hand context. We have both left and right contexts available to examine because the Brill tagger makes a first pass through the data to assign its best guess to each word in the text. Then we break the text into training pairs for the neural network. The details of the approach are recorded in [3] and [4], but this paper represents the first results to be published.

The five output nodes for each neural network represent the five part-of-speech categories that the network could suggest for the target word (noun, verb, adjective, adverb, and gerund). These five were chosen because they represent the categories with which main verbs were most often confused by the Brill tagger.

0-7803-4859-1/98 $10.00©1998 IEEE

Num. of Hidden Units	L2 R0 Error	L4 R0 Error	L6 R0 Error	L8 R0 Error	L10 R0 Error	L2 R2 Error	L4 R2 Error	L6 R2 Error	L6 R4 Error
300	0.175	0.174	0.182	0.193	0.190	0.126	0.129	0.133	0.132
500	0.165	0.183	0.186	0.192	0.196	0.118	0.134	0.134	0.141

Table One: Fully Connected, One Output Bit High, Best Training Error Rate Achieved

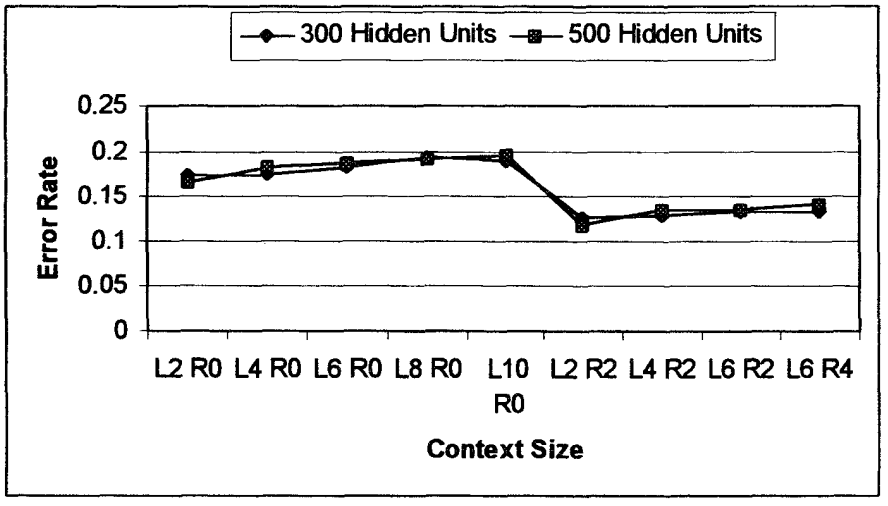

Figure One: Graph of data from Table One

Only the highest valued output bit would be considered "high" for comparison to the expected "high" output bit. As explained in [3] and [4], the networks were trained on nine files of training pairs, and then validated on a tenth file which had been held out. No back-propagation was performed while using the validation file. Table One shows the lowest error rate reported on the validation file during training of the networks.

As Figure One makes clear, the fully connected networks performed comparably. Note that extremely long context sizes do not horribly affect the ability of the network to recognize patterns. Also, networks which include data from both the left and right contexts perform better than using only left context (statistics-based taggers use only the left context).

Based on the algorithm proposed in [9], we implemented a search for fractally-configured neural networks and ran a second set of experiments. We did modify the algorithm slightly, and this is reported in [10]. Since some context sizes seemed to be relatively less productive in the fully connected networks, we did not continue with the same set of contexts. We no longer examine the L2 R0, the L8 R0, and the L10 R0 contexts. Instead, we have added an L6 R6 context, which means that there are six part-of-speech context tags on each side of the target word. Table Two shows the lowest error rate achieved on the validation file for a fractally configured network discovered during the search process outlined in [9]. These networks are in the same error percentage range as the fully connected networks, but have fewer weight connections due to the fractal edge configuration. A similar architecture might be developed by pruning a fully connected network after it was trained, but Merrill and Port [9] argue that pruning after training is semantically different from training a network that is only partially connected in the beginning. They further argue that a fractal configuration will improve performance over a fully connected network because the mere existence and absence of links encodes domain knowledge for the neural network.

The best network architectures discovered during the search process for fractal configuration were further trained using standard back-propagation for 120 epochs, where one epoch consists of training once on all the pairs in the nine training files. The error rate for these networks on the validation file (again, no back-prop) is shown in Table Three. As above, note that networks with some right-hand part-of-speech context are better able to

Num. Of Hidden Units	L4 R0 Error	L6 R0 Error	L2 R2 Error	L4 R2 Error	L6 R2 Error	L6 R4 Error	L6 R6 Error
300	Not avail	0.242	Not avail	0.139	0.142	0.142	0.149
500	0.186	0.182	0.142	0.149	0.153	0.147	0.154

Table Two: Fractally Connected, One Output Bit High, Best Search Error Rate Achieved

Num. Of Hidden Units	L4 R0 Error	L6 R0 Error	L2 R2 Error	L4 R2 Error	L6 R2 Error	L6 R4 Error	L6 R6 Error
500	0.180	0.184	0.127	0.129	0.129	0.134	0.146

Table Three: Fractally Connected, One Output Bit High, Best Training Error

identify the complex patterns and suggest a correct part-of-speech to the target word. Note also that these training error results are not terribly different from the results obtained by the fully connected networks in Table One.

Conclusion

In this paper, we have compared two types of neural network architecture as applied to the problem of assigning a part-of-speech tag to a word based on recognizing a complex and extended context around the target word. Fully connected networks and fractally configured networks perform approximately the same on this task. Previous research had shown that around 300 hidden units were needed to perform adequately on a similar but more simple task, and we show here that adding further hidden units does not significantly improve the performance on a larger task. Since a network with 500 hidden units consumes quite a few more resources (both in terms of memory and number of operations required for each training pair), there doesn't seem to be justification for pursuing large networks on this real-world task. In addition, the resources required to carry out the search algorithm for a fractal configuration are significant and the results do not seem to justify the expense in that there is no significant improvement in performance. The good news is that we were able to make a 12.9% improvement in the number of mis-labelings made on main verbs, which is a particularly damaging tagging error.

References

[1] Boggess, J. E. and Lois Boggess. 1994. A hybrid probabilistic/connectionist approach to automatic text tagging. In *Proceedings of the 7th Florida Artificial Intelligence Research Symposium*, Pensacola Beach, Florida, pp. 147-51.

[2] Boggess, Lois, Julia Hodges, and Jose Cordova. 1995. Automated knowledge derivation: Domain-independent techniques for domain-restricted text sources. *International Journal of Intelligent Systems*, **10**(10), pp. 871-93.

[3] Boggess, Lois and Lynellen D. S. Perry. 1997. Real world auto-tagging of scientific text. In *Proceedings of the 10th International Florida Artificial Intelligence Research Symposium*, Daytona Beach, Florida, May 12-14, pp. 253-7.

[4] Boggess, Lois and Lynellen D. S. P. Smith. 1996. But 'propeller' is a verb! Automatic tagging and noun/verb confusions. In *Proceedings of the 9th Florida Artificial Intelligence Research Symposium*, Key West, Florida, May 20-22, pp. 511-5.

[5] Brill, Eric. 1992. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*. Association for Computational Linguistics, pp. 152-5.

[6] Brill, Eric. 1994. Some advances in transformation-based part of speech tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence*. Menlo Park: AAAI Press/MIT Press, pp. 722-7.

[7] Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, **21**(4):543-65.

[8] Hodges, Julia, Shiyun Yie, Sonal Kulkarni, and Ray Reighart. 1997. Generation and evaluation of indexes for chemistry articles. *Journal of Intelligent Information Systems* **8**(1): 57-76.

[9] Merrill, John W., and Robert F. Port. 1991. Fractally configured neural networks. *Neural Networks*, **4**, pp. 53-60.

[10] Perry, Lynellen D. S. 1997. Improving the identification of verbs. In *Proceedings of the 35th Annual Southeast Conference of the ACM*, Murfreesboro, Tennessee, April 2-4, pp. 41-3.