

Using Machine Learning Methods to Improve Quality of Tagged Corpora and Learning Models

Yuji Matsumoto and Tatsuo Yamashita

Graduate School of Information Science
Nara Institute Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0101, Japan
{matsu,tatuo-y}@is.aist-nara.ac.jp

Abstract

Corpus-based learning methods for natural language processing now provide a consistent way to achieve systems with good performance. A number of statistical learning models have been proposed and are used in most of the tasks which used to be handled by rule-based systems. When the learning systems come to such a level as competitive as manually constructed systems, both large scale training corpora and good learning models are of great importance. In this paper, we first discuss that the main hindrances to the improvement of corpus-based learning systems are the inconsistencies or the errors existing in the training corpus and the defectiveness in the learning model. We then show that some machine learning methods are useful for effective identification of the erroneous source in the training corpus. Finally, we discuss how the various types of errors should be coped with so as to improve the learning environments.

1. Introduction

Using corpora to improve natural language processing systems has now become an established technique. Part-of-speech (POS) tagging and syntactic dependency analyzers are typical examples, in which statistical technique achieves competitive performance against human-crafted systems. Although statistical techniques relying only on unannotated plain corpora are well-known, such as HMM (Hidden Markov Model) learning (Huang et al., 1990; Jelinek, 1998) for POS tagging and Inside-Outside algorithm for probabilistic context-free grammars (Charniak, 1993), annotated corpora are especially of great importance. No matter how large they are, unannotated corpora hardly qualify learning systems to achieve an accuracy as high as those that learn on an annotated corpus of moderate size.

Recently, a number of supervised learning methods have been proposed (Manning & Schütze, 1999), which with an annotated corpus can build Natural Language Processing systems with a very good performance. Even a simple trigram-based POS tagger of English achieves an accuracy well over 95%, and some sophisticated mixture models such as those using bagging or boosting can enhance the performance into non-significant level (Haruno & Matsumoto, 1997). When learning methods evolve to such a level, the accuracy of annotated corpora themselves should not be underestimated. It is actually quite difficult to ensure consistency in annotated corpora. In our experience, the agreement rate of two trained human annotators working on Japanese POS tagging is initially less than 93%. This means that careful screening of annotated data is inevitable. Now, the major hindrance of our current statistical learning tasks for POS tagging and syntactic dependency analysis is the existence of errors or inconsistencies in the training corpus.

Another difficulty in corpus-based learning systems is the model selection. Even though an accurately annotated training corpus is available, a learning system cannot achieve high performance if the learning model it bases on is not sophisticated enough. Moreover, even if the learning

model is sophisticated, we still need to know the way to find out the set of features that enables the learning system to attain better performance.

In this paper, we show that recent machine learning methods are useful not only to construct systems of high performance but also to identify corpus errors effectively and to get the hints for selecting good learning models.

2. Types of Errors in Statistical Language Processing

We have been developing a statistical system for tokenization, morphological analysis and POS tagging, called ChaSen (Matsumoto et al., 1999), and a statistical dependency parser (Fujio & Matsumoto, 1998). An experimental system for language independent tokenization and POS tagging has been implemented (Yamashita & Matsumoto 2000), and ChaSen incorporates the idea and runs both on Japanese and English texts.

Those systems learn their statistical parameters from annotated corpora. When we examine the errors committed by the learned system, we found that the causes of errors can be classified into five types as follows:

1. Simple tagging errors that remain in the training corpus.
2. Set of tags that do not have clear boundaries.
3. Unseen/rare events missing in the training corpus.
4. Lack of features or defectiveness of the learning model.
5. Errors that cannot be perfectly solved by the employed learning model.

We will discuss those causes of errors in detail in the next subsections.

2.1. Simple tagging errors

There may be a number of errors that have been tagged incorrectly or simply been overlooked by the annotators. This type of errors are not the fault of the learned system but the fault of the corpus annotators. This type of errors should be corrected and removed from the training corpus anyhow.

2.2. Unclear boundaries among ambiguous tags

There are a number of critical pairs of POS tags that are hard to distinguish even by a well-trained annotator. For example, distinction among prepositions (IN), particles (RP) and adverbs (RB) is not an easy task. Though there is a guideline for such confusing cases (Santorini, 1990), it is unable to list all the misleading cases and to describe the ways for discrimination in the guideline.

In the case of Japanese or Chinese, in which word boundaries are not segmented by separators such as white spaces, there are ambiguities in the segmentation of words. Compound words are the source of segmentation ambiguity. For example, a word like “ComputerScienceLaboratory” may be regarded as a single proper noun or a compound word consisting of three words (or four words if a suffix is regarded as an independent word).

2.3. Rare events

Some type of errors are caused by the sparseness of data. The test data may have a chance to encounter some events that never or rarely occur in the training corpus. Then, appropriate learning of probabilistic values has not been performed. This situation happens quite often when the test data is of different domain from the training corpus.

This type of errors is easily remedied by adding the missing examples to the training corpus. However, modification of the learning models or application of some smoothing methods can be more direct solution.

2.4. Lack of features or defectiveness of the learning model

Some errors cannot be corrected without taking into consideration a more fine-grained feature set, in case the granularity of the model is not fine enough. Some requires a refinement of the model and the other requires addition of new features. For example, our Japanese POS tagger ChaSen uses mixture of bigram and trigram rules. Full trigram model cannot be learned since the tag set is quite large (about 500 different tags when all the inflection forms for verbs, adjectives and auxiliary verbs are counted). The system used to use bigram model. Then, some of the distinction is impossible since they are affected by the words appearing at some distance from the ambiguous words. The current ChaSen system employs a variable-length N-gram model and accepts contexts of unlimited length. At the moment bigram learning is mainly used and trigram rules are selectively employed. A grouping technique of words or POS tags is also allowed to refine the base model and to cope with data sparseness problem. The details of them and other extension will be reported in (Asahara & Matsumoto 2000). Some linguistic phenomena that cannot be

captured by bigram rules require some trigram extension, and a modification of the model become necessary to cope with the errors caused by such phenomena.

2.5. Errors out of the scope of the model

The cause of some ambiguity may go beyond the description facility of the learning model. For example, there are a number of tagging ambiguities that require information far away from a limited window size. For example, Japanese postpositional particles have a number of grammatical functions and categorized into several different subclasses. When a POS tagger tries to discriminate such grammatically distinct usages, it requires a number of information which simple bigram or trigram-based systems are unable to have access to. Although correct tagging to such data is important, the learning model should better not commit to such fine-grained distinction of tags.

We believe it is quite important to identify and locate errors in the training corpus for statistical Natural Language Processing systems to achieve higher performance as well as for us to maintain consistency of tagged corpora. In the next section, we show that a machine learning technique is usable to find out those errors existing in tagged corpora.

3. Use of Machine Learning Method for Error Identification

We employ the boosting algorithm for our current purpose of detecting errors. We once showed that mixture of models based on boosting improves the performance of a POS tagger (Haruno & Matsumoto, 1997). Here, however, we first see that the same idea is useful to locate errors and inconsistencies in the corpus quite effectively. We employ the AdaBoost algorithm (Freund & Shapire, 1996) in this paper. The sketch of the AdaBoost algorithm is shown in Figure 1.

At each iteration of the algorithm, a new system is trained by the original corpus with different weight to the words. Then, the final system is constructed as a weighted mixture of the whole learned systems where the weights for the systems are decided by the error rate of themselves.

Here, we are not interested in the learned system. We are more interested in the final weights given to the words in the training corpus. The important feature of the AdaBoost algorithm is that at each iteration the learner puts greater focus on the erroneous parts in the data. The larger weight a word has, the harder it has been for the learner to assign the correct tag. Actually, the tagging error corresponds to erroneous or problematic parts in the annotate corpus, and the use of AdaBoost is a good way to pin down such parts.

We applied five iteration of the AdaBoost procedure with a trigram-based learning of POS tagging using the POS tagged version of Penn TreeBank (Marcus et al., 1993). After five iteration, 14% of the word tokens get a higher weight than the lowest one, meaning at least one learner has failed to tag the tokens correctly. 2.5% of the whole tokens are tagged wrongly by at least two learners.

From the latter tokens we randomly pick up 100 tokens and examined them manually.

47 out of them are annotated with incorrect tags, mainly of type 1 errors. 46 are correct. 4 are difficult to classify

Step1: Let the size of the training data be N . Every word in the given annotated corpus is assumed to have a uniform distribution (a uniform weight, $\forall i.weight[i] = \frac{1}{N}$).

Step2: Do the following for $t = 1, 2, \dots, T$:

1. Call the statistical learning program to train the POS tagger using the corpus taking into account the weight vector given by the current distribution.
2. Evaluate the error rate ϵ_t of the learned system h_t by the following formula (*Error* means the set of words that are tagged wrongly by h_t):

$$\epsilon_t = \sum_{w_i \in Error} \frac{weight[i]}{\sum_{j=1}^N weight[j]}$$

3. For each word wrongly tagged by the learned system, put an extra-weight to the word according to the error rate as follows:

$$weight[i] = weight[i] \times \frac{\epsilon_t}{1 - \epsilon_t}$$

Then, recalculate the distribution of the weight vector using the new weights.

Figure 1: Flow of AdaBoost Algorithm

due to the delicate distinction of preposition(IN) vs particle(RP) and adverb(RB) vs particle(RP), etc, which are categorized as type 2 or 3 errors in the previous section. Finally, 3 are cases of compound nouns, compound proper nouns and combination of them, to which simple trigram model cannot fully cope with. Those are categorized into type 2 or 4 errors.

Although the experiment is very primitive at the moment, almost 50% error identification rate is quite high, suggesting that this method is very effective in finding errors. While half of the erroneous parts are not strictly errors, the reason they get high weights is that there are some source of difficulties that hinder the correct tagging to the data. The real causes vary the types of the reasons shown in the preceding section.

Of course, any learning system can be used to detect erroneous or difficult parts in the training data simply analyzing the data by the learned system and comparing the results with the original data. We used to take this approach to correct errors in manually tagged corpus. We showed in this section, the method like boosting, which focuses attention on relatively difficult fragments of data, gives more effective means to identify real errors.

4. Coping with the Causes of Errors

We discussed that the analysis errors caused by corpus-based learning system can be categorized into five types. Then, in the preceding section we showed that the erroneous parts can be detected easily by using AdaBoost algorithm. There are variety of causes of errors as discussed in

Section 2. This section extends the discussion to the consideration of how to cope with those different types of errors.

Types 1 and 5 are the extreme cases: For the former case, where the analysis errors are actually caused by annotation errors in the training corpus, they should be corrected immediately. For the latter case, where the errors are caused by difficult phenomena that cannot be fully solved by the currently employed model, we should leave them out of consideration. In our case of Japanese tagged corpus, after applying the cycles of training the POS tagger and correcting the tagged corpus, most of the errors of the highest frequencies have become such errors.

We will discuss the means to cope with other types of errors in more detail.

4.1. Tagged corpus as the guideline

Type 2 errors are caused by the inconsistency that remains in the tagged corpus. As discussed in Section 2, distinction between prepositions (IN) and particles (RP) is a difficult task. Compiling a tagging guideline helps annotators only to some extent, since there are always exceptions and it is tiresome for annotators to consult the guideline every now and then.

Without a good guideline, annotators are always in danger of making mistakes in tagging some confusing samples. To cope with such situations, we rather do not try to write detailed guidelines, but make use of *tagged corpus as the guideline*.

We developed a KWIC (Key Words In Context) system, in which a tagged corpus can be retrieved by any information from word form (inflected or base form) to part-of-speech name. The system shows the retrieved results as KWIC format. Then the user can specify further (contextual) conditions to limit the retrieved results. The user may take statistical summaries of the retrieved words or of the surrounding words. By using this system, the user can consult the tagged corpus in various ways and take statistics on the fly.

For example, when an annotator is confused by the distinction between DT and PDT for the word ‘all,’ he/she first retrieves the word by issuing two queries, ‘all/DT’ and ‘all/PDT,’ then compares the results by looking at two results in KWIC format. A tendency may be recognized that every occurrence of ‘all/PDT’ is followed by another determiner. If the annotator likes to ensure the guess, he/she can take the statistics of the surrounding words. Taking the statistics of part-of-speech tags of the words appearing to the right of ‘all/PDT’ reveals that all of those words are determiners (DT’s). On the other hand, the statistics for ‘all/DT’ shows several alternative parts-of-speech at the same position, but with no occurrence of determiners. From this the annotator learns the distinction of DT and PDT.

By letting annotators use this system, we found that the idea of *tagged corpus as the guideline* is a quite effective method. The idea shows that a tagged corpus could replace a detailed description of tagging guideline. The system is also useful for novice annotators to learn the POS tag set.

Figures 2 and 3 show sample snapshots of the KWIC system running on a Japanese tagged corpus. Figure 2 is the main window showing the usages of light verb “suru”

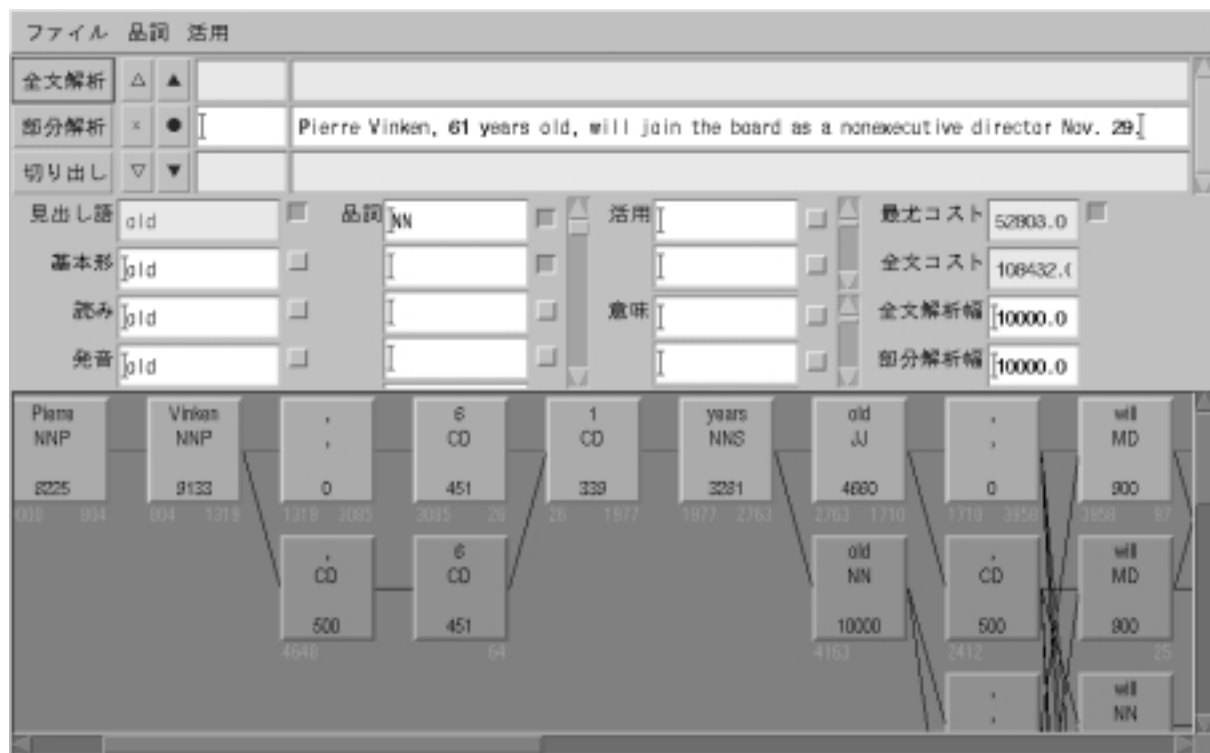


Figure 4: VisualMorphs: Corpus annotation environment

4.3. Refinement of learning model

Type 4 errors come from immaturity of the learning model. Refinement of learning models should be considered in two directions. One is the width or the range of the model, meaning how far the contextual information is taken into consideration. For instance, trigram models have wider range than bigram models. The other direction of refinement of models is the depth or the grain size of information used in the model. Word-level statistics may be more useful than POS-level statistics as far as the data size is enough. It is not always possible to use word-level statistics in the face of limited size of training data. We always have to consider the trade-off between the refinement of the model and data sparseness problem.

Taking those issues into consideration, we are working on extension of learning models and learning tools for POS tagging. For the problem of range of the learning model it employs a variable-length model, and for grain-size it employs an extended model of grouping POS tags. The details are found in (Asahara & Matsumoto 2000).

5. Related Work and Further Research

It is natural that machine learning techniques are unable to find inconsistencies or errors in the training data, since the errors by the learned system indicate at least some anomalies exist in the data or in the learner. Abney et al.,(1999) describes that boosting is useful for identification of errors in annotated corpus. We classified the causes of errors by the learned system into five types and discussed how to cope with them. The advantage of Boosting to the tasks introduce in the paper is its generalization power. As

it is called a high margin classifier, the probability of incidental mis-classification tends to be small so that the erroneous data tend to be more informative than just using simple learner. Another candidate of high margin classifier is Support Vector Machine (SVM) (Vapnik, 1998). Further investigation of SVM-based identification of error source may be an interesting topic.

We are applying the method to our Japanese POS tagger ChaSen and hopefully to statistical dependency parser. ChaSen now employs a variable memory length Markov model and accepts contexts of any length. The current system mainly use a full bigram model with specially selected trigram rules. By effectively identifying the errors that require further information over bigram models, we will attain efficient selection of useful trigram instances. Automatic or semi-automatic refinement of learning models both for the width and depth is the aim of further research.

6. Conclusions

Though corpus-based learning methods for natural language processing are now widely used, accumulation of annotated corpora and refinement of learning models are constant demands. In this paper, we discussed that the main obstacles to further improvement of corpus-based learning systems are the inconsistencies or the errors existing in the training corpus and the defectiveness in the learning model. We then showed that machine learning methods such as boosting are useful for effective identification of the erroneous parts in the training corpus. We also discussed how the various types of errors should be coped with to improve the learning systems.

Acknowledgements

The authors would like to express sincere thanks to Akira Kitauchi, Yoshitaka Hirano, Hiroshi Matsuda, Masayuki Asahara and many other members of Computational Linguistics Laboratory in Nara Institute of Science and Technology (NAIST), who helped to develop the Japanese morphological analyzer, ChaSen, and many other systems such as corpus browser/annotator and statistical learning programs.

7. References

- Abney, S., Schapire, R.E. and Singer, Y., 1999. Boosting Applied Tagging and PP Attachment *Proc. Joint SIG-DAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 38–45.
- Asahara, M. and Matsumoto, Y., 2000. Extended Models and Tools for High-performance Part-of-speech Tagger. to appear in *Proc. 18th International Conference on Computational Linguistics*.
- Charniak, E., 1993. *Statistical Language Learning*. The MIT Press.
- Engelson, S.P. and Dagan, I., 1996. Sample Selection in Natural Language Learning. Wermter, S. et al.(eds.), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing, Lecture Notes in Artificial Intelligence 1040*. Springer, 230–245.
- Freund, Y. and Shapire, R., 1996. Experiments with a New Boosting Algorithm. *Proc. 13th International Conference on Machine Learning*. 148–156.
- Fujio, M. and Matsumoto, Y., 1998. Japanese Dependency Structure Analysis based on Lexicalized Statistics. *Proc. 3rd Empirical Methods in Natural Language Processing*, 88–96.
- Haruno, M. and Matsumoto, Y., 1997. Mistake-Driven Mixture of Hierarchical Tag Context Trees. *Proc. 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of Computational Linguistics*. 230–237.
- Huang, X.D., Ariki, Y. and Jack, M.A., 1990. *Hidden Markov Models for Speech Recognition*. Edinburgh Information Technology Series (EDITS) 7, Edinburgh University Press.
- Jelinek, F., 1998. *Statistical Methods for Speech Recognition*. The MIT Press.
- Manning, C.D. and Schütze, H., 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.
- Marcus, M.P., Santorini, B., Marcinkiewicz, M.A., 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*. 19(2):313–330.
- Matsumoto, Y., Kitauchi, A., Yamashita, T., Hirano, Y., Matsuda, H. and Asahara, M., 1999. *Japanese Morphological Analysis System ChaSen 2.0 Users Manual, 2nd edition*. Technical Report NAIST-IS-TR99012, Nara Institute of Science and Technology.
- Santorini, B., 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project. *Technical Report, University of Pennsylvania*.

- Yamashita, T. and Matsumoto, Y., 2000. Language Independent Morphological Analysis. *Proc. 6th Conference on Applied Natural Language Processing*.
- Vapnik, V.N., 1998. *Statistical Learning Theory*. John Wiley & Sons.