Mitigating the Paucity-of-Data Problem: Exploring the Effect of Training Corpus Size on Classifier Performance for Natural Language Processing

Michele Banko and Eric Brill
Microsoft Research
1 Microsoft Way
Redmond, WA 98052 USA
{mbanko, brill}@microsoft.com

ABSTRACT
In this paper, we discuss experiments applying machine learning techniques to the task of confusion set disambiguation, using three orders of magnitude more training data than has previously been used for any disambiguation-in-string-context problem. In an attempt to determine when current learning methods will cease to benefit from additional training data, we analyze residual errors made by learners when issues of sparse data have been significantly mitigated. Finally, in the context of our results, we discuss possible directions for the empirical natural language research community.

Keywords
Learning curves, data scaling, very large corpora, natural language disambiguation.

1. INTRODUCTION
A significant amount of work in empirical natural language processing involves developing and refining machine learning techniques to automatically extract linguistic knowledge from on-line text corpora. While the number of learning variants for various problems has been increasing, the size of training sets such learning algorithms use has remained essentially unchanged. For instance, for the much-studied problems of part of speech tagging, base noun phrase labeling and parsing, the Penn Treebank, first released in 1992, remains the de facto training corpus. The average training corpus size reported in papers published in the ACL-sponsored *Workshop on Very Large Corpora* was essentially unchanged from the 1995 proceedings to the 2000 proceedings. While the amount of available on-line text has been growing at an amazing rate over the last five years (by some estimations, there are currently over 500 billion readily accessible words on the web), the size of training corpora used by our field has remained static.

Confusable word set disambiguation, the problem of choosing the correct use of a word given a set of words with which it is commonly confused, (e.g. {to, too, two}, {your, you're}), is a prototypical problem in NLP. At some level, this task is identical to many other natural language problems, including word sense disambiguation, determining lexical features such as pronoun case and determiner number for machine translation, part of speech tagging, named entity labeling, spelling correction, and some formulations of skeletal parsing. All of these problems involve disambiguating from a relatively small set of tokens based upon a string context. Of these disambiguation problems, lexical confusables possess the fortunate property that supervised training data is free, since the differences between members of a confusion set are surface-apparent within a set of well-written text.

To date, all of the papers published on the topic of confusion set disambiguation have used training sets for supervised learning of less than one million words. The same is true for most if not all of the other disambiguation-in-string-context problems. In this paper we explore what happens when significantly larger training corpora are used. Our results suggest that it may make sense for the field to concentrate considerably more effort into enlarging our training corpora and addressing scalability issues, rather than continuing to explore different learning methods applied to the relatively small extant training corpora.

2. PREVIOUS WORK
2.1 Confusion Set Disambiguation
Several methods have been presented for confusion set disambiguation. The more recent set of techniques includes multiplicative weight-update algorithms [4], latent semantic analysis [7], transformation-based learning [8], differential grammars [10], decision lists [12], and a variety of Bayesian classifiers [2,3,5]. In all of these papers, the problem is formulated as follows: Given a specific confusion set (e.g. {to, two, too}), all occurrences of confusion set members in the test set are replaced by some marker. Then everywhere the system sees this marker, it must decide which member of the confusion set to choose. Most learners that have been applied to this problem use as features the words and part of speech tags

appearing within a fixed window, as well as collocations surrounding the ambiguity site; these are essentially the same features as those used for the other disambiguation-in-string-context problems.

2.2 Learning Curves for NLP

A number of learning curve studies have been carried out for different natural language tasks. Ratnaparkhi [12] shows a learning curve for maximum-entropy parsing, for up to roughly one million words of training data; performance appears to be asymptoting when most of the training set is used. Henderson [6] showed similar results across a collection of parsers.

Figure 1 shows a learning curve we generated for our task of word-confusable disambiguation, in which we plot test classification accuracy as a function of training corpus size using a version of winnow, the best-performing learner reported to date for this well-studied task [4]. This curve was generated by training on successive portions of the 1-million word Brown corpus and then testing on 1-million words of Wall Street Journal text for performance averaged over 10 confusion sets. The curve might lead one to believe that only minor gains are to be had by increasing the size of training corpora past 1 million words.

While all of these studies indicate that there is likely some (but perhaps limited) performance benefit to be obtained from increasing training set size, they have been carried out only on relatively small training corpora. The potential impact to be felt by increasing the amount of training data by any signifcant order has yet to be studied.

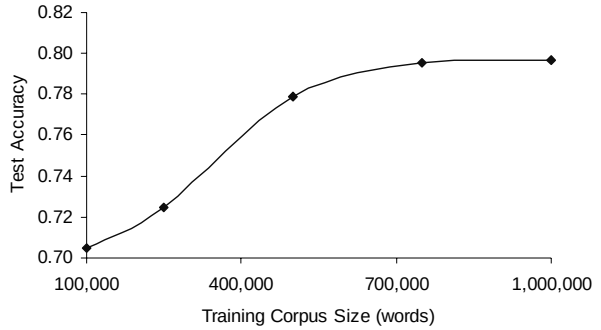

Figure 1: An Initial Learning Curve for Confusable Disambiguation

3. EXPERIMENTS

This work attempts to address two questions – at what point will learners cease to benefit from additional data, and what is the nature of the errors which remain at that point. The first question impacts how best to devote resources in order to improve natural language technology. If there is still much to be gained from additional data, we should think hard about ways to effectively increase the available training data for problems of interest. The second question allows us to study failures due to inherent weaknesses in learning methods and features rather than failures due to insufficient data.

Since annotated training data is essentially free for the problem of confusion set disambiguation, we decided to explore learning curves for this problem for various machine learning algorithms, and then analyze residual errors when the learners are trained on all available data. The learners we used were memory-based learning, winnow, perceptron,[1] transformation-based learning, and decision trees. All learners used identical features[2] and were used out-of-the-box, with no parameter tuning. Since our point is not to compare learners we have refrained from identifying the learners in the results below.

We collected a 1-billion-word training corpus from a variety of English texts, including news articles, scientific abstracts, government transcripts, literature and other varied forms of prose. Using this collection, which is three orders of magnitude greater than the largest training corpus previously used for this task, we trained the five learners and tested on a set of 1 million words of Wall Street Journal text.[3]

In Figure 2 we show learning curves for each learner, for up to one billion words of training data.[4] Each point in the graph reflects the average performance of a learner over ten different confusion sets which are listed in Table 1. Interestingly, even out to a billion words, the curves appear to be log-linear. Note that the worst learner trained on approximately 20 million words outperforms the best learner trained on 1 million words. We see that for the problem of confusable disambiguation, none of our learners is close to asymptoting in performance when trained on the one million word training corpus commonly employed within the field.

Table 1: Confusion Sets

{accept, except}	{principal, principle}
{affect, effect}	{then, than}
{among, between}	{their, there}
{its, it's}	{weather, whether}
{peace, piece}	{your, you're}

The graph in Figure 2 demonstrates that for word confusables, we can build a system that considerably outperforms the current best results using an incredibly simplistic learner with just slightly more training data. In the graph, Learner 1 corresponds to a trivial memory-based learner. This learner simply keeps track of all $<w_{i-1}, w_{i+1}>$, $< w_{i-1}>$ and $<w_{i+1}>$ counts for all occurrences of the confusables in the training set. Given a test set instance, the learner will first check if it has seen $<w_{i-1},w_{i+1}>$ in the training set. If so, it chooses the confusable word most frequently observed with this tuple. Otherwise, the learner backs off to check for the frequency of $<w_{i-1}>$; if this also was not seen then it will back off to $<w_{i+1}>$, and lastly, to the most frequently observed confusion-

[1] Thanks to Dan Roth for making both Winnow and Perceptron available.

[2] We used the standard feature set for this problem. For details see [4].

[3] The training set contained no text from WSJ.

[4] Learner 5 could not be run on more than 100 million words of training data.

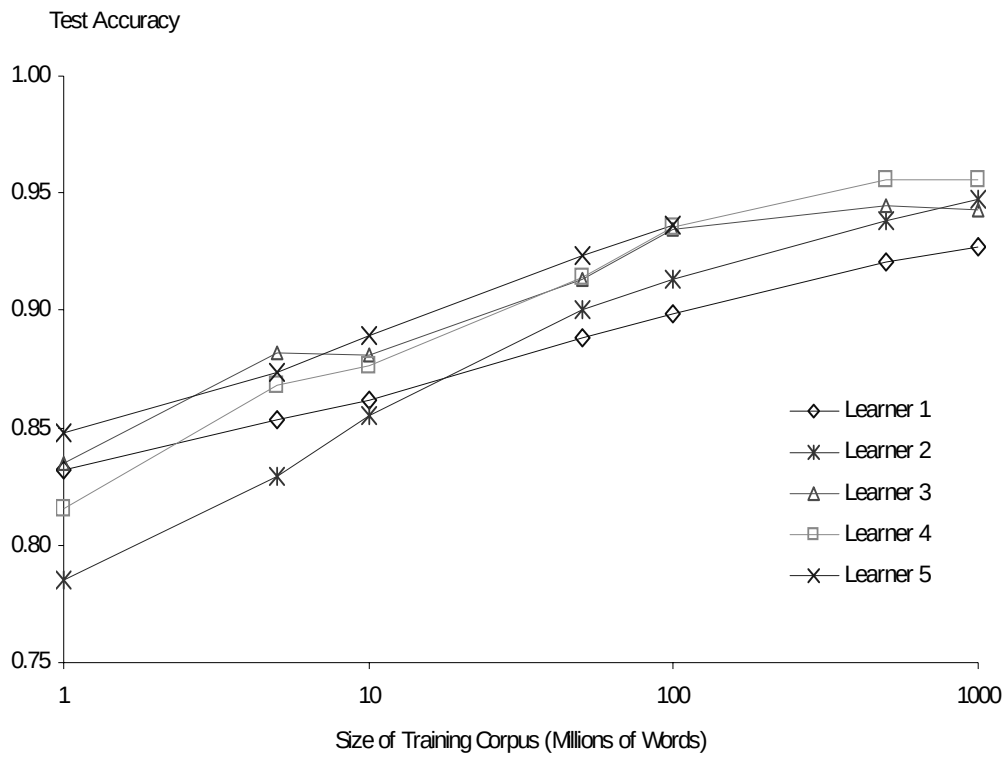

Figure 2. Learning Curves for Confusable Disambiguation

set member as computed from the training corpus. Note that with 10 million words of training data, this simple learner outperforms all other learners trained on 1 million words.

Many papers in empirical natural language processing involve showing that a particular system (only slightly) outperforms others on one of the popular standard tasks. These comparisons are made from very small training corpora, typically less than a million words. We have no reason to believe that any comparative conclusions drawn on one million words will hold when we finally scale up to larger training corpora. For instance, our simple memory based learner, which appears to be among the best performers at a million words, is the worst performer at a billion. The learner that performs the worst on a million words of training data significantly improves with more data.

Of course, we are fortunate in that labeled training data is easy to locate for confusion set disambiguation. For many natural language tasks, clearly this will not be the case. This reality has sparked interest in methods for combining supervised and unsupervised learning as a way to utilize the relatively small amount of available annotated data along with much larger collections of unannotated data [1,9]. However, it is as yet unclear whether these methods are effective other than in cases where we have relatively small amounts of annotated data available.

4. RESIDUAL ERRORS

After eliminating errors arising from sparse data and examining the residual errors the learners make when trained on a billion words, we can begin to understand inherent weaknesses in ourlearning algorithms and feature sets. Sparse data problems can always be reduced by buying additional data; the remaining problems truly require technological advances to resolve them.

We manually examined a sample of errors classifiers made when trained on one billion words and classified them into one of four categories: strongly misleading features, ambiguous context, sparse context and corpus error. In the paragraphs that follow, we define the various error types, and discuss what problems remain even after a substantial decrease in the number of errors attributed to the problem of sparse data.

Strongly Misleading Features

Errors arising from *strongly misleading features* occur when features which are strongly associated with one class appear in the context of another. For instance, in attempting to characterize the feature set of *weather* (vs. its commonly-confused set member *whether*), according to the canonical feature space used for this problem we typically expect terms associated with atmospheric conditions, temperature or natural phenomena to favor use of *weather* as opposed to *whether*. Below is an example which illustrates that such strong cues are not always sufficient to accurately disambiguate between these confusables. In such cases, a method for better weighing features based upon their syntactic context, as opposed to using a simple bag-of-words model, may be needed.

Example: *On a sunny day <u>whether</u> she swims or not depends on the temperature of the water.*

Ambiguous Context
Errors can also arise from *ambiguous contexts*. Such errors are made when feature sets derived from shallow local contexts are not sufficient to disambiguate among members of a confusable set. Long-range, complex dependencies, deep semantic understanding or pragmatics may be required in order to draw a distinction among classes. Included in this class of problems are so-called "garden-path" sentences, in which ambiguity causes an incorrect parse of the sentence to be internally constructed by the reader until a certain indicator forces a revision of the sentence structure.

Example 1: *It's like you're king of the hill.*

Example 2: *The transportation and distribution departments evaluate weather reports at least four times a day to determine if delivery schedules should be modified.*

Sparse Context
Errors can also be a result of *sparse contexts*. In such cases, an informative term appears, but the term was not seen in the training corpus. Sparse contexts differ from ambiguous contexts in that with more data, such cases are potentially solvable using the current feature set. Sparse context problems may also be lessened by attributing informative lexical features to a word via clustering or other analysis.

Example: *It's baseball's only team-owned spring training site.*

Corpus Error
Corpus errors are attributed to cases in which the test corpus contains an incorrect use of a confusable word, resulting in incorrectly evaluating the classification made by a learner. In a well-edited test corpus such as the Wall Street Journal, errors of this nature will be minimal.

Example: *If they don't find oil, its going to be quite a letdown.*

Table 2 shows the distribution of error types found after learning with a 1-billion-word corpus. Specifically, the sample of errors studied included instances that one particular learner, winnow, incorrectly classified when trained on one billion words. It is interesting that more than half of the errors were attributed to sparse context. Such errors could potentially be corrected were the learner to be trained on an even larger training corpus, or if other methods such as clustering were used.

The ambiguous context errors are cases in which the feature space currently utilized by the learners is not sufficient for disambiguation; hence, simply adding more data will not help.

Table 2: Distribution of Error Types

Error Type	Percent Observed
Ambiguous Context	42%
Sparse Context	57%
Misleading Features	0%
Corpus Error	1%

5. A BILLION-WORD TREEBANK?

Our experiments demonstrate that for confusion set disambiguation, system performance improves with more data, up to at least one billion words. Is it feasible to think of ever having a billion-word Treebank to use as training material for tagging, parsing, named entity recognition, and other applications? Perhaps not, but let us run through some numbers.

To be concrete, assume we want a billion words annotated with part of speech tags at the same level of accuracy as the original million word corpus.[5] If we train a tagger on the existing corpus, the naïve approach would be to have a person look at every single tag in the corpus, decide whether it is correct, and make a change if it is not. In the extreme, this means somebody has to look at one billion tags. Assume our automatic tagger has an accuracy of 95% and that with reasonable tools, a person can verify at the rate of 5 seconds per tag and correct at the rate of 15 seconds per tag. This works out to an average of 5*.95 + 15*.05 = 5.5 seconds spent per tag, for a total of 1.5 million hours to tag a billion words. Assuming the human tagger incurs a cost of $10/hour, and assuming the annotation takes place after startup costs due to development of an annotation system have been accounted for, we are faced with $15 million in labor costs. Given the cost and labor requirements, this clearly is not feasible. But now assume that we could do perfect error identification, using sample selection techniques. In other words, we could first run a tagger over the billion-word corpus and using sample selection, identify all and only the errors made by the tagger. If the tagger is 95% accurate, we now only have to examine 5% of the corpus, at a correction cost of 15 seconds per tag. This would reduce the labor cost to $2 million for tagging a billion words. Next, assume we had a way of clustering errors such that correcting one tag on average had the effect of correcting 10. This reduces the total labor cost to $200k to annotate a billion words, or $20k to annotate 100 million. Suppose we are off by an order of magnitude; then with the proper technology in place it might cost $200k in labor to annotate 100 million additional words.

As a result of the hypothetical analysis above, it is not absolutely infeasible to think about manually annotating significantly larger corpora. Given the clear benefit of additional annotated data, we should think seriously about developing tools and algorithms that would allow us to efficiently annotate orders of magnitude more data than what is currently available.

6. CONCLUSIONS

We have presented learning curves for a particular natural language disambiguation problem, confusion set disambiguation, training with more than a thousand times more data than had previously been used for this problem. We were able significantly reduce the error rate, compared to the best system trained on the standard training set size, simply by adding more training data.

[5] We assume an annotated corpus such as the Penn Treebank already exists, and our task is to significantly grow it. Therefore, we are only taking into account the marginal cost of additional annotated data, not start-up costs such as style manual design.

We see that even out to a billion words the learners continue to benefit from additional training data.

It is worth exploring next whether emphasizing the acquisition of larger training corpora might be the easiest route to improved performance for other natural language problems as well.

7. REFERENCES

[1] Brill, E. Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging. In Natural Language Processing Using Very Large Corpora, 1999.

[2] Gale, W. A., Church, K. W., and Yarowsky, D. (1993). A method for disambiguating word senses in a large corpus. Computers and the Humanities, 26:415--439.

[3] Golding, A. R. (1995). A Bayesian hybrid method for context-sensitive spelling correction. In Proc. 3rd Workshop on Very Large Corpora, Boston, MA.

[4] Golding, A. R. and Roth, D. (1999), A Winnow-Based Approach to Context-Sensitive Spelling Correction. Machine Learning, 34:107--130.

[5] Golding, A. R. and Schabes, Y. (1996). Combining trigram-based and feature-based methods for context-sensitive spelling correction. In Proc. 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA.

[6] Henderson, J. Exploiting Diversity for Natural Language Parsing. PhD thesis, Johns Hopkins University, August 1999.

[7] Jones, M. P. and Martin, J. H. (1997). Contextual spelling correction using latent semantic analysis. In Proc. 5th Conference on Applied Natural Language Processing, Washington, DC.

[8] Mangu, L. and Brill, E. (1997). Automatic rule acquisition for spelling correction. In Proc. 14th International Conference on Machine Learning. Morgan Kaufmann.

[9] Nigam, K, McCallum, A, Thrun, S and Mitchell, T. Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning. 39(2/3). pp. 103-134. 2000.

[10] Powers, D. (1997). Learning and application of differential grammars. In Proc. Meeting of the ACL Special Interest Group in Natural Language Learning, Madrid.

[11] Ratnaparkhi, Adwait. (1999) Learning to Parse Natural Language with Maximum Entropy Models. Machine Learning, 34, 151-175.

[12] Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In Proc. 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM.