An investigation into compositional features and feature merging for maximum entropy-based parse selection
Mullen, Anthony James

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2002

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
Chapter 6

Overfitting reduction with feature merging for models with compositional features

6.1 The Alvey Tools experiments

This chapter describes experiments in maximum entropy-based parse selection using parses generated by the Alvey Tools Tag Sequence Grammar.

Several models are described, using various feature sets. The problem of overfitting is described and several approaches to reducing it are presented, including the novel approach of feature merging. It is shown that a combination of approaches is most effective at improving a model which suffers from overfitting.

6.1.1 How this work differs from other work in the field

When viewed in contrast with the work in statistical parsing discussed in chapter 2, there are some significant differences in the intentions and the methodology of the present work. Whereas those efforts were task specific and aimed at maximizing the recognition of syntactic constraints, the present work seeks to investigate strategies of optimizing the results gained by very general methods. The grammar used in the experiments in this chapter is a general purpose, wide coverage grammar developed independently of any specific application and (more importantly) without regard for any specific data set. Likewise, the grammar of Dutch examined in chapter 7 is intended as a general purpose grammar. The maximum entropy method of modeling is general to any statistical modeling problem, and in fact had applications in fields as diverse as physics and image processing long before it was employed in NLP. These differences in intention and approach mean that the numbers reported in this work will necessarily find an upper bound considerably lower than the state of the art for specific tasks un-
dertaken using specifically developed systems. This is unfortunate, as it makes it somewhat difficult to evaluate the results presented here in light of other work in the field. Furthermore, difficulties arise in finding absolute precision and recall scores for the selected parses due to the necessity of translating parses in the grammar format into parses in the Penn Treebank format. This is painstaking work and for the present purposes its investigative and expository usefulness is questionable. The results reported in this chapter for various models are internally comparable, and since the focus is on methods of improving the models, this is sufficient for the present work. Later in this chapter I will say more about the difficulties of evaluation this work in absolute terms.

6.2 The problem of overfitting

As discussed in earlier chapters, the maximum entropy technique of statistical modeling has proved to be an effective way of dealing with a variety of linguistic phenomena. This is largely because its capacity for considering overlapping information sources allows the most to be made of situations where data is sparse. Nevertheless, it is important that the statistical features employed be appropriate to the job. If the information contributed by the features is insufficiently general, overfitting becomes a problem (Chen and Rosenfeld, 1999; Osborne, 2000a). On the other hand, underfitting is the inability of a model to make distinctions due to the overly general nature of its features.

Simply put, overfitting refers to the tendency of a model to conform to aspects of the training data which are not generally applicable to unseen data; the model fits the training (seen) data too well to generalize accurately and thus its performance on test data suffers. This occurs when there is noise in the training data in the form of errors, or when some piece of data occurs too rarely in the training data for its frequency to be accurately reflected. In such cases, the performance of the model suffers. Often, overfitting will manifest itself in a drop in performance after a certain number of training iterations; as the sample more accurately estimates the training data, its performance on unseen data worsens. This phenomenon is referred to as overtraining. In this case, a peak in model performance will be reached early on, and continued training yields progressive deterioration in performance. In other cases, the noisy aspects of the training data may be modeled early on in the training process. From a theoretical standpoint, overfitting indicates that the model distribution is unrepresentative of the actual probabilities. In practice, overtraining can be avoided by ending training early, although the point at which this must be done is not always reliably predictable. The more general problem of overfitting, however, including the more subtle manifestation which does not necessarily change the shape of the curve over training iterations but simply results in diminished performance all along the curve, can most effectively be reduced by eliminating sources of noise within
the model itself.

In the maximum entropy framework, statistics are collected for features. These features may be of various degrees of specificity, and thus may tend to be more or less frequent in the training data. It would be possible to define a feature, for example, which was true any time a noun part of speech occurred in the data. Such a feature would be very general and frequent, whereas a feature which was defined as true given the presence of the lexical trigram “about the ptarmigan” would be vanishingly rare. Usually, overfitting is associated with a feature set where the information contributed by the features tends towards the overly specific.

6.3 Parse selection with compositional features

The work described in this chapter follows upon work presented in Mullen (2000) and Mullen and Osborne (2000). An approach to feature selection for maximum entropy models is introduced in which candidate features are built up from more basic elements found in the corpus, as described in chapter 5. This “compositional” quality of the features is exploited for the purpose of overfitting reduction by means of feature merging. In this process, features which are similar to each other, aside from certain elements, are merged; i.e., their disjunction is considered as a feature in itself, thus reducing the number of features in the model. As with approaches to feature subset selection such as Della Pietra, Della Pietra, and Lafferty (1995) and Kohavi and John (1997), the motivation here is to arrive at a smaller set of features which is better able to represent underlying frequencies, by means of discarding information sources in the model which contribute noise and hinder accurate generalization. In these approaches, however, the information discarded is in the form of “irrelevant” features (Kohavi and John, 1997), whereas the present approach considers the possibility of compressing the feature set through merging, rather than seeking a proper subset of the feature set. The method differs likewise from a simple feature cutoff, such as that described in Ratnaparkhi (1998), in that the feature cutoff eliminates statistical features directly, whereas the merging procedure attempts to generalize them. The method employed here also derives some inspiration from the notion of Bayesian model merging introduced by Stolcke and Omohundro (1994), although the notion there applied to HMM models is very different from the present notion of merging compositional statistical features.

The task of parse selection involves selecting the best possible parse for a sentence from a set of possible parses produced by a grammar. In the present approach, parses are ranked according to their goodness by a statistical model built using the maximum entropy technique. The maximum entropy principle states that in cases where no information is present in the training data which would support a preference for one distribution to another, the distribution should be
uniform, i.e., the entropy of the distribution should be maximal. Following this guideline, the maximum entropy technique involves building a distribution over events which is the most uniform possible, given constraints derived from empirical frequencies in the training data. That is, the distribution should be the unique distribution which conforms to the empirical frequencies of the data, while being otherwise uniform. It is the distribution which has the maximum entropy of all distributions which conform to the empirical frequencies of features, the fundamental statistical units of which events are composed, and whose distribution is modeled. The constraints which characterize the model are expressed as weights on individual features. Training the model involves deriving the best weights from the training data by means of an algorithm such as the IIS algorithm introduced in chapter 3.

IIS assigns weights to features which reflect their distribution and significance. With each iteration, these weights reflect the empirical distribution of the features in the training data with increasing accuracy. In ideal circumstances, where the distribution of features in the training data accurately represents the true probability of the features, the performance of the model should increase asymptotically with each iteration of training until training ceases to reduce the KL divergence (see section 3.2.4) between the model distribution and the empirical distribution. If the training data is corrupt, or noisy, or if it contains features which are too sparsely distributed to accurately represent their probability, then overfitting arises.

6.4 Feature merging and overfitting reduction

The idea behind feature merging is to reduce overfitting through changes made directly to the model. This is done by combining highly specific features which occur rarely to produce more general features which occur more often, resulting in fewer total features used. Even if the events are not noisy or inaccurate in actual fact, they may still contribute to overfitting if their features occur too infrequently in the data to give accurate frequencies. The merging procedure seeks to address overfitting at the level of the features themselves and remain true to the spirit of the maximum entropy approach, which seeks to represent what is unknown about the data with uniformity of the distribution, rather than by making adjustments on the model distribution itself, such as the Gaussian prior of Chen and Rosenfeld (1999).

Each feature, as described above, is made up of discrete elements, which may include such objects as lexical items, POS tags, and grammatical attribute information, depending on the schema being used, as discussed in section 5.4.1. The infrequency of the feature in the data is largely—although not entirely—determined by the infrequency of elements within it. In the present merging scheme, a set of elements is collected whose empirical frequencies are below some
Figure 6.1: An example of feature merging. The features are of the variety described in section 5.4.1. The top two features are merged in the form of the bottom feature, where the lexical elements have been replaced by their disjunction. The merged feature represents the union of the sets of tokens described by the unmerged feature types. All instances of the original two features would now be replaced in the data by the merged feature.
predetermined cutoff point. Note that the use of the term "cutoff" here refers to the empirical frequency of elements of features rather than of features themselves, as in Ratnaparkhi (1998). All features containing elements in this set will be altered such that the cutoff element is replaced by a uniform disjunctive element, effectively merging all similarly structured features into one, with the disparate elements replaced by the disjunctive element. An example may be seen in figure 6.1, where the union of the two features at top of the figure is represented as the feature below them. The merged elements in this case are the lexical items offered and allow. Such a merge would take place on the condition that the empirical frequencies of both elements are below a certain cutoff point. This number is referred to as the "merging threshold." If the counts of both elements are lower than the merging threshold, the elements are replaced by a new element representing the disjunction of the original elements, creating a single feature. This feature then replaces all instances of both of the original features. If both of the original features appear once each together in an event, then two instances of the merged feature will appear in that event in the new model.

One insight gained in the following experiments and those of chapter 7 is that in cases where overfitting is a significant problem in the model, the merging process can be expected to improve performance if circumstances in the model are, by and large, analogous to those presented in the toy example in section 5.5. That is to say, if the merged features are, in fact, the closest approximation to accurate backed-off features available to the model, we would expect to see concrete improvements in performance. As will be observed in chapter 7 and discussed in chapter 8, this is not always the case.

6.5 Experiments with Alvey Tools

The experiments described here were conducted using the Wall Street Journal Penn Treebank corpus (Marcus, Santorini, and Marcinkiewicz, 1993; Marcus et al., 1994). The grammar used was a manually written broad coverage DCG style grammar, the Alvey Tools Tag Sequence Grammar (version tsg11) (Briscoe and Carroll, 1997). A description of the parser used can be found in Carroll (1993). Following Osborne (2000a), parses of WSJ sentences produced by the grammar were ranked empirically using the treebank parse as a gold standard according to a weighted linear combination of crossing brackets and shared constituents (Hektoen, 1997), specifically: $x + y - 3z$ where

- $x =$ the number of shared constituents
- $y =$ the number of shared constituents with matching labels
- $z =$ the number of crossing constituents

If more than fifty parses were produced for a sentence, the best fifty were used and the rest discarded. For the training data, the empirical rankings of all parses for each sentence were normalized so the total parse scores for each sentence
added to a constant, as was described in section 5.3. The events of the training
data consisted of parses, and the frequency of each event was determined by
the corresponding normalized score. Thus, highly ranked parses were treated as
events occurring more frequently in the training data, and low ranked parses were
treated as occurring rarely, following the approach taken by Osborne (2000a).

The features were derived in the manner described in section 5.4.1. The fea-
tures of the unmerged model consisted of depth-one trees carrying node infor-
mation according to the following schema: the syntactic category of the mother,
POS tags of all daughters ordered left to right, HEAD± information for the head
daughter, and lexical information for all daughters carrying a verbal or prepo-
tional POS tag. The features themselves were culled using this schema on 2290
sentences from the training data. The feature set consisted of 38,056 features in
total, of which 4,832 were active in the model (assigned non-zero weights) follow-
ing the last iteration of H1S. Two models using this feature set were trained, one
on only 498 training sentences, a subset of the 2290 sentences used to collect the
features, and the other on the full 2290 sentences.

Several merged models were made based on each of these unmerged models,
using various cutoff numbers, ranging from less than 100 to 10,000. For each
model merge, all elements which occurred in the training data fewer times than
the cutoff number were replaced in each feature they appeared in by the uniform
disjunctive element, and the merged features then took the place of the unmerged
features.

Iterative scaling was performed for 150 iterations on each model. This number
was chosen arbitrarily as a generous but not gratuitous number of iterations,
allowing general trends to be observed. Optimal cutoffs and merging thresholds
were derived from a held-out test set of approximately 5,000 unseen sentences,
and another set, about the same size, was used as the actual test set.

6.5.1 Evaluation

The performance of each model was measured at each iteration by binary best
match. The model chose a single top parse and if this parse's empirical rank was
the highest (or equal to the highest) of all the parses for the sentence, the model
was awarded a point for the match, otherwise the model was awarded zero. The
performance rating reflects the percentage of times that the model chose the best
parse of all possible parses, averaged over all test sentences. In some cases, there
were ties, both in the scores assigned by the model and in the empirical scores.
In these cases the following approach was taken: in cases of ties in the empirical
scores, an exact match point was awarded if the model selected any of the top-
scored parses; in cases where the model selected more than one parse with equal
probability, one of these parses was chosen at random to be the selected parse. In
this way, ties were neither unduly punished nor unduly rewarded. The baseline
was determined by evaluating randomly selected parses according to the same
exact match criteria.

This exact match approach to evaluation allows the salient results of this research to be clearly seen. It does not however reflect the absolute performance of the grammar and statistical model together in parsing the Wall Street Journal. My own efforts at translating grammar parses into Penn Treebank style parses so that they might be compared to yield meaningful PARSEVAL scores yielded disappointing results. The top ranked grammar parses averaged only 67 recall and less than 40 precision when compared to the Penn Treebank parses. This should not be taken as an accurate reflection of the grammar’s potential with regard to the Penn Treebank, but only as an indication that the translation function between the two styles of trees was inadequate. Since this measure was not immediately pertinent to conveying the results of the present experiments, but rather only to evaluating the upper bound presented by the grammar, I did not pursue it further.

6.6 Results

6.6.1 Performance of unmerged models

Of the unmerged models the one trained on the smaller set shows the worst performance and most drastic overfitting. Its peak at around 36.5% performance comes early, at around 10 iterations of HH, and subsequently drops gradually to around 35% by around 100 iterations. This model’s performance may be seen in figure 6.2 represented by the solid black line. The dotted line represents a random model, used to obtain a baseline measurement.

6.6.2 Performance of merged models

Different thresholds yielded varying degrees of improvement. A variance of 100 elements seemed to make no meaningful difference either way with either model; the best balance seemed to be struck with a merging threshold of 200.

As can be seen from figure 6.3, the merged model, represented by the dotted line, shows some improvement over the original model. The reduction of noise has increased the peak performance to just under 38%, although with steady training the downward subsequent curve remains.

1Readers familiar with the publications Mullen (2000) and Mullen and Osborne (2000) will notice that the graphs shown here are not the same as those presented in those papers. In fact, the results presented in those papers were in error, due to bugs in the software (written by myself) used for experimentation and evaluation. The errors resulted in inflated performance scores, but also obscured the improvements brought about by the merging technique. The present results are the product of extensive debugging and re-testing. I would like to apologize for the earlier errors.
Figure 6.2: The original model, trained on 498 sentences.
Figure 6.3: For the model trained on 498 sentences, features containing elements appearing fewer than 2000 times are merged.
6.7. FREQUENCY BASED FEATURE CUTOFFS

Figure 6.4: The model with a cutoff of 150 only.

6.7 Frequency based feature cutoffs

A common practical approach to overfitting reduction is the use of frequency based feature cutoffs (Ratnaparkhi, 1998).

Figure 6.4 shows the improvement gained by eliminating all features which appear fewer than 150 times. This number was derived using a held-out test set. This cutoff improves the shape of the curve significantly, but the peak performance does not reach the level of that in the merged model.

6.7.1 Problems with feature cutoffs

Cutting off features works to reduce overfitting by discarding noisy information. The smallest unit of information which can be discarded is that contained in a single feature itself. One of the strengths of maximum entropy modeling, as has been discussed, however, is the possibility of arbitrarily complex features, which can represent overlapping sources of information. For elimination of noise, therefore, it would appear to be desirable, when possible, to attempt to eliminate noisy elements within features, while keeping as much information as possible
Figure 6.5: The cutoff model and the merged model compared.
from the feature to use in modeling. This is the motivation behind the merging approach introduced in section 6.4. The two methods of noise reduction are by no means mutually exclusive. As can be seen from the comparison of the two models in figure 6.5, one with the cutoff and the other with merging, the resulting models, while neither is clearly better than the other, are quite different. This is due to the fact that both methods of reducing overfitting target different sources of noise. Feature merging eliminates specific noise resulting from noisy elements which cannot be eliminated by using a straight feature cutoff (without also eliminating other, non-noisy information) whereas noisy features which are not simply the result of rare elements can not be eliminated by use of the merging approach. This occurs when elements which are not sparse in the data are arranged within the feature in a way which is highly uncommon.

6.8  A combined approach

In order to derive the greatest benefit from both methods of noise reduction, a balance should be struck between merging and the use of cutoffs. The method employed in these experiments was to merge the model first, according to a predetermined threshold, and perform the cutoff on the resultant feature set. The merging pass reduced the number of infrequent features by merging many infrequent features together to create features which occurred less infrequently; the frequency of a merged feature, of course, being the sum of frequencies of all those features which have been unioned to create it. On the cutoff pass, then, fewer infrequent features remain to be discarded. Those which were below the cutoff due to infrequent elements have been merged, so that their remaining information is still used by the model. The resultant model is considerably smaller than the original model, containing only 911 active features.

In cases such as the model discussed so far, where overfitting and noise is a major problem, the combination of approaches results in significantly improved performance, as can be seen in figure 6.6.

Figure 6.6 shows how the use of both merges and a subsequent feature cutoff yields superior results to either method on its own.

6.9  Increased training data and richer feature sets

The methodologies discussed in this chapter and elsewhere in this thesis are primarily concerned with manipulations done directly on a given feature set to increase its performance through noise reduction. Of course, depending on available resources, there are other ways to improve the performance of models. Increasing the amount of training data available to the model significantly reduces
Figure 6.6: A combined approach, using both merging and cutoff.
the problem of sparse data, yielding reliable frequencies for features which are too rare in small data sets to accurately predict.

Increasing the amount of training data to 2290 sentences improved the results considerably and also nearly eliminated the improvement which could be gained by cutoffs and merges. Figure 6.7 shows the best improvement that was reached using a cutoff. No amount of prior merges was found to improve upon this result. Although the performance increase is not particularly impressive, it is worth pointing out that the model size has been reduced greatly. The original model, trained on 2290 sentences, had 16331 active features, while the cutoff model had only 3522 active features. Prior merging with a threshold of 500 decreased this number a little further, to 3349, with no apparent decrease in performance.

### 6.9.1 Backing off with compositional features

Another method for attaining better generalization is to begin with a richer model. The use of compositional features built up of elements such as those described in chapter 5 allows feature sets of varying richness to be described in a straightforward manner, exploiting the ability of the maxent technique to deal well with overlapping information sources. The addition of “backed off” features to the model improved the performance even more. It is notable, however, that this approach reduces overfitting by increasing the size of the model, whereas the merging/cutoff method does so by decreasing the size of the model. In this case features were added which consisted solely of the mother’s category and bar information and the daughters POS tags. The total number of active features in this model, trained on 498 sentences was 9776, around twice that of the non-backed off model, and on 2290 training sentences, the total number of active features was 33928, also just over twice the number of features in the previous model. The addition of such backed off features does not reduce the presence of noisy features in the data, of course, but rather reduces the significance of the noisy features. The impact of the noise on the outcome of the selection process is thus reduced, and performance increases. With these features added to a model trained on 498 sentences, IIS stopped (the KL divergence ceased to decrease) after only 5 iterations. The performance at this stage was .457. With a training set of 2290, this same feature set yielded the best performance of these experiments, reaching about 53.5% performance immediately after five iterations and not diverging from this by more than around 0.5% for the duration of training. No combination of cutoff or merges was found to improve upon the performance of this model.
CHAPTER 6. IMPROVED MODELS WITH FEATURE MERGING

Figure 6.7: Five times the training data, with and without cutoff of 50.
Figure 6.8: A richer feature set with more training data yields the best results, with no evidence of overfitting.
6.10 Conclusion

Both the merging technique and the frequency-based feature cutoff eliminate noise from a model. The experiments using the Alvey Tools grammar presented in this chapter demonstrate that under certain circumstances, given a feature set with certain qualities, the merging technique can improve the performance of a model, as seen in figure 6.3. The merged model does not itself outperform a model with a cutoff applied; the two are different but roughly comparable, as can be seen in figure 6.5. However, when applied in advance of the cutoff, the merging can yield a model which outperforms either technique on its own, as can be seen in figure 6.6.

It is important to recognize, however, that this is not the case for all the models looked at in this chapter. The model for which the merging/cutoff combination did most good was, unfortunately, the worst performing model initially, and trained on only a small portion of the available training data. The merging was not found to have any positive effect at any threshold, and even a small cutoff made little clear difference. When a richer feature set was introduced, whose performance is seen in figure 6.8, any cutoff or merging threshold appeared only to worsen performance.

The reason for this once again related to the ability of the maxent technique to take into consideration overlapping information sources and to weight features according to how informative they are. The richer feature set is distinguished from the others primarily in its use of backed-off features, which, due to their greater frequency, provide more accurate, although less specific, information about their probability than the more specific features of the other models. In this case, the importance of noisy features is automatically diminished, thus eliminating noise makes little difference. Any merge threshold or cutoff large enough to impact the model is necessarily too large not to impact it negatively.

6.10.1 Why investigate further?

Ideally, it is best to have a rich, comprehensive feature set with both general and specific features, and to have a sufficient amount of representative training data to make overfitting reduction unnecessary. Unfortunately, it is not always possible to have enough data available to ensure that overfitting is not a problem. The investigation into the possibility of feature merging was initially inspired as a means of reducing the size of the feature set and reducing noise through compression. This remains a worthwhile pursuit, and in particular, it appears likely to be worthwhile in cases where small training data sets lead to serious sparse data problems.

It seems plausible that in any situation where a straight frequency-based feature cutoff is helpful, merging might be used to improve matters even more. This is because any time a feature cutoff is employed, information is lost which is
not noisy, simply because it occurs in the same feature as noisy information. The question, however, is how great the benefit is. It is possible that, given backed-off features and a reasonable amount of training data, the improvement gained by merging prior to a cutoff is negligible.

In chapter 7, the techniques discussed in this chapter are adapted to deal with the problem of parse selection in the Dutch Alpino grammar environment. The basic idea of analyzing the features as compositional and merging based upon common elements within them is the same. This environment is a good testing ground for the approach, since training corpus is quite small. In the Alvey Tools experiments in this chapter, the models which showed improvement were deliberately impoverished. In the Alpino environment, resources are in fact much more limited, and it is hoped that the techniques investigated here can in fact improve upon the best performance attained so far.
CHAPTER 6. IMPROVED MODELS WITH FEATURE MERGING