

Chapter 6

Conclusion

Exempting the first two chapters that introduce the reader to the history, theory, and implementation of ILP systems, this thesis has been concerned with the extension of an ILP system to allow for the evaluation of clauses in parallel (Chapter 3) and two applications of ILP; one on Syntax (Chapter 4) and one on Phonology (Chapter 5).

This concluding chapter will first summarise and discuss the computational cost issues encountered during these experiments (Section 6.1). Then the results and conclusions of the two experimental chapters will be brought together and summed up in Section 6.2. The points made there will be further explored in the discussion that follows in Section 6.3, where the experience of applying ILP to these linguistic tasks will be discussed and reflected upon.

The thesis closes by suggesting *future directions* for research in applying ILP for linguistic tasks in Section 6.4 — the inevitable admission that this work is not complete, since there are always more paths to be explored and ideas to be tried.

6.1 Computational Complexity

As already noted in the introduction of Chapter 3, ILP runs are notoriously slow and memory-intensive, which makes parallel computation a very attractive prospect. The attempt at a data-parallel version of Aleph described in that chapter did not, however, yield very promising results, demonstrating that the bottleneck is the size of the search space — that is, the number of clauses that need to be constructed and evaluated — rather than the cost of the evaluation phase per se.

What makes the problem even more acute is the observation made in Section 5.7 about the growth rate of the search space as a function of the size of

the bottom clause. Since the more complex, long and useful the background theory is, the longer a bottom clause it will produce, one can immediately see that interesting background theories will make data-parallelism even less attractive an option for ILP.

One thing that should be noted at this point is that the measurements for the data-parallel Aleph system were carried out on a workstation cluster rather than a parallel machine. Workstation clusters are arrays of workstations communicating via a network connection and not sharing memory, meaning that they suffer from a high communication overhead, by comparison to parallel machines. Furthermore, broadcasting — on which Aleph/MPI heavily relies — is especially affected by the fact that there is no shared memory from which all receiving nodes can copy the broadcast message, but a number of point-to-point connections has to be established instead.

On the other hand, clusters have to offer larger numbers of nodes, since a network is much more easily expendable than a parallel machine. This makes them more appropriate for highly parallelised tasks with little communication among nodes and near-linear benefits from each node added to the computation. Aleph/MPI on the other hand is not such an application: (a) there is regular communication between the nodes, which although not very voluminous does require that the overhead be repeatedly paid, and (b) Ahmdal's Law takes effect rather quickly, since a large part of the computation remains serial. This implies that Aleph/MPI would benefit more from a shared-memory multi-processor machine with low communication overhead and few nodes, than from a cluster with high communication costs and a much larger number of nodes.

6.2 Summation of Results

The evaluation of the ILP experiments conducted in Chapters 4 and 5 has been concerned with the quantitative as well as the qualitative analysis of the resulting theories. The quantitative results from the application of ILP to capture linguistic phenomena are encouraging, although not spectacular. In both cases very high compression is achieved: BaseNP chunking is, effectively, done with just 11 clauses (see Section 4.5) whereas the phonotactic rules of Dutch can be expressed with 181 or 106 clauses, depending on the informedness of the background theory (see Section 5.8).

But although ILP achieves high compression of the data, suggesting high generalization power, the precision and recall rates for the BaseNP chunking experiment are inferior to the competition: the ILP-induced chunker achieved a recall rate of 85.32% with 78.62% precision, which compares poorly with

System	ML Approach	Precision	Recall
Kudoh and Matsumoto	SVM (*)	93.45%	93.51%
Van Halteren	WPDV/MBL (*)	93.13%	93.51%
Tjong Kim Sang	MBL (*)	94.04%	91.00%
Zhou, Tey and Su	HMM	91.99%	92.25%
Déjean	ALLiS	91.87%	92.31%
Koeling	MaxEnt	92.08%	91.86%

Table 6.1: Performance of top chunkers from the CoNLL-2000 Shared Task. The starred (*) approaches reach a decision by combining the results of multiple learners of the type(s) shown on the table.

the top results from other approaches on the slightly more complicated task of BaseXP recognition, as can be seen from Table 6.1, the results of the CoNLL-2000 Shared Task [82]. (References to the papers describing each system also available *ibidem*.)

On the other hand, in the domain of Dutch phonotactics the ILP approach described in Chapter 5 shows considerably improved results by comparison to an earlier abductively induced theory [83] on both performance and compression (see Section 5.8). More specifically, the experiment with the simpler background (IPA table, Section 5.4) slightly improves upon the performance of the abductive theory with considerably fewer clauses. Furthermore, the more informed background (Booij’s Feature Classes, Section 5.5) gives rise to a theory that is even more compact and considerably more precise.

The qualitative analysis of the resulting theories has also shown chunking (or at least the experimental setup chosen) to be ill-suited for applying ILP, since the promise of readability and conciseness of the resulting theory and the benefits from the usage of background knowledge — advertised as ILP’s biggest advantages both in Section 2.6 and in the literature in general — are not fully substantiated. As already noted in the concluding section of the relevant chapter (Section 4.5), the word-tagging setup used distributes each chunking decision over a number of individual word-tagging decisions, which are to a large extent made independently. This makes it difficult to interpret the theory, since the chunks end up depending a lot on the interaction between rules. It also renders the background knowledge less useful, since some framework constraints (such as, for example, those imposed by X-bar theory) are not expressible in this setup.

A similar approach was chosen for the phonotactics experiments described in Chapter 5: the setup was such that the recognizer would make individual decisions about attaching a phoneme to an already recognized and accepted

‘kernel.’ In this case, however, this approach yielded much more interesting results: the resulting theories were more concise as well as more accurate than the abductively constructed ones, as shown in Section 5.8.

The promising result of the qualitative analysis of the chunking experiment is that what little background knowledge is available does get used in a meaningful way: a common pattern is clauses where the *naive/2* background predicate is used to assign tags to words, with the other literals in the clause restricting its domain of application. Similarly in the phonotactics experiments, the quality and sophistication of the background knowledge has an impact on the performance and size of the theory, also suggesting that the information encoded in the background does get used and does make a difference.

6.3 Discussion

It is more interesting to analyse the results of the ILP experiments in conjunction with the results of other systems on the same task, and a comparison of ILP with other machine learning approaches tackling the BaseXP chunking task was given above.

By examining the best performing systems (see Table 6.1 above), we find three systems combining the results of multiple chunkers to reach a decision at the top three positions: the Kudoh-Matsumoto system takes a majority-vote decision based the results of the complete set of pairwise classifiers; Weighted Probability Distribution Voting (WPDV) for Van Halteren; and majority voting for Tjong Kim Sang. Kudoh and Matsumoto later [38] improved the performance of their system even further, by adding another layer of voting and applying weighted-voting among 8 systems like the ones used for the CoNLL-2000 Shared Task.

Furthermore, the individual chunkers the decision is based on, are based themselves on numerical or stochastic Machine Learning algorithms¹ for the top two systems, and only at the third position appears a majority-voting combination of (symbolic) Memory-Based Learners. The single-chunker systems trail with Hidden Markov Models (Zhou, Tey and Su), followed by a symbolic theory-refinement system (ALLiS, employed by Déjean), and then the procession of non-symbolic approaches goes on with Maximum Entropy learners and Markov Models, and only at the very last places symbolic approaches appear again. Notice how all chunker combinations out-perform

¹Support Vector Machines for Kudoh-Matsumoto and Weighted Probability Distribution Voting for Van Halteren, although the latter also includes one Memory-Based Learner among the five chunkers used.

all single-chunker systems, and within each of these two classes stochastic and numerical systems out-perform the symbolic ones, with the exception of ALLiS.

It seems, then, that systems that combine the results of multiple learners are better suited for the task, especially when they are combining the results of non-symbolic machine learning systems, and that ILP is simply the wrong tool for this kind of problem. Reflecting on the general properties of ILP, one notices that fuzzy and noisy concepts pose the biggest problems for the induction of logic programmes, and for formal logic in general. BaseNP chunking is such a noisy domain, since quite often the same string of part-of-speech tags will be chunked differently, depending on semantic or long-distance syntactic features that are outside the scope of the chunker. This is, naturally, not implying anything with regards to using logic formalisms in NLP in general, but only in tasks like chunking where the processing is performed on limited information so as to keep the processing fast and simple.

One can also see that the ILP learner suffered from the noise of the data in the large number of ungeneralized examples remaining at the end, versus the small number of general clauses successfully induced: eleven out of 160, as noted in Section 4.5. Relaxing the accuracy requirements would improve this ratio, but only at the expense of precision, yielding over-generalizing clauses.

Furthermore, as has already been noted, only a very limited amount of background knowledge was made available to the learning process, due to the very nature of the experiment's setup. This setup has become quite standard in the literature since the seminal work of Ramshaw and Marcus [69], but it seems to be ill-suited for the purposes of ILP. On the other hand, a more complex formalism — like the CFGs suggested by Abney [1] — would constitute an unjustified² overkill, since finite-state theories have been successfully induced for this problem. From the above we can conclude that chunking is a problem that is difficult in the wrong way, so that it does not benefit from the advantages of ILP but it does bring out its weaknesses.

And then again, a similar setup in the domain of phonotactics has yielded much better results. This can be attributed to the following factors:

- Phonotactic data is much less noisy: native Dutch words can be described consistently, with all the relevant features available to the learner. That is, no information³ was excluded from the background, so all the elements of the decision are available. This leaves only relatively new loan-words that have not been phonologically dutchified as the sole

²computational complexity-wise

³As, for example, semantics was excluded for the chunking experiment.

source of noise, and those were appropriately treated as exceptional singularities.

- Phonology (unlike syntax) is an inherently local phenomenon, so that the prior knowledge and the theoretical framework is also more readily expressible in a local, finite-state formalism. In other words, no prior knowledge is wasted due to the difficulty or impossibility of expressing it within the experimental setup. The generalization power of ILP is, then, employed to compress the number of affix rules necessary to fully describe the phenomenon.

Schaffer [73] showed that a *conservation law* must hold for the generalization performance of each particular learning system. That is, it is not possible for a particular system to perform well on all domains, but performance increase in one dataset will have to be balanced by lower performance in some other datasets. In other words, the kind of generalizations a system is looking for will be appropriate for some problems, but not for others. This result is, of course, simply adding some extra theoretical support to well known maxims of Computer Science, such as *use the right tool for the job* because *there's no such thing as a free lunch!*

6.4 Future Directions

The discussion above provides some pointers towards interesting ways to follow up on the work described in this thesis, which should be read as complementing (rather than summarising) the partial future research suggestions at the end of each chapter.

Due to the highly disjunctive nature of predicate definitions in Logic Programming, ILP is well suited to capture the outlying data-points that remain after a numerical or stochastic system has explained the main body of the data. That is, of course, true of other symbolic systems as well. In fact, a combination system like Van Halteren's (see above) would rely on its Memory Based Learning component to do exactly that: to detect the exceptional cases that would either lie outside the Weighted Probability Distribution Voting theories or cause them to over-generalize.

It would, therefore, be interesting to see how well an ILP-generated chunker complements statistical chunkers in a combination system like the one of Van Halteren. This would provide data for a comparison between ILP and Memory Based Learning on the task of complementing a battery of stochastic systems.

In the light of the gains, in terms of results, achieved by improving the background theory in the phonotactics experiments, it would be interesting to experiment further with the linguistic knowledge in the chunking experiment as well. The linguistic background provided there was restricted to features like `NOMINAL` or `VERBAL`, which simply hinted at the syntactic properties of each part-of-speech, but provided no information other than various linguistically motivated ways to partition the set of part-of-speech tags. This proved much more effective in the phonotactics experiments, but — as already noted in the discussion above — phonotactics is an inherently local phenomenon and syntax is not.

What then emerges as an interesting possibility, is some experimentation which breaks with the long chunking-as-word-tagging, finite-state tradition that stems from the original Ramshaw and Marcus experiment. To be able to capitalize better on the strong points of ILP, the setup should be such that long-distance dependencies should be discouraged⁴ but nevertheless explored, and kept if found to be making a distinction that cannot be otherwise made. Access to a dictionary (or, in any case, a more complex lexeme ontology than a simple part-of-speech tag) would also be something that would be easily incorporate-able as prior in ILP.

⁴In order to retain the fast-and-shallow character of chunking, and not gravitate towards a full parser.

