7 Conclusion

In this final chapter, I will firstly consider the results of this study in broad terms. I will discuss this in terms of the corpus annotation tools I have developed (7.1.1), in terms of the project's duration (7.1.2), and in terms of the discoveries I have made about the Urdu language (7.1.3). I will consider the success that I have had, the aspects of the project that have not been successful, and improvements which could have been made. I will then move on to discuss possible future avenues of research into tagging Urdu (7.2). I will then look back over this thesis and provide a summary of the claims I have made and the aims I have fulfilled (7.3).

7.1 Results of this study

7.1.1 Resources developed

Two important resources have been developed for the future study of Urdu corpus linguistics: the tagging system itself, and the tagset which it applies.

7.1.1.1 The tagset

The U0 tagset (see Chapter 3) and the U1 and U2 subtagsets (see Chapter 4) obviously represent a major resource for Urdu corpus linguistics. It is not only a key part of the tagger presented here. Its value as analysis scheme is independent of any particular application, and as such it is a useful product of this study in its own right.

Of course, the analysis scheme as a whole is greater than just the tagset. The

tagging guidelines which explain how the tagset is to be implemented are vitally necessary if the analysis scheme is to be consistent, as explained in Chapter 4. Therefore, the tagging guidelines too are an important output of this study.

Furthermore, there was linguistic value in the process of creating the tagset. I was able to demonstrate (in Chapter 3) the extensibility of the EAGLES guidelines on morphosyntactic annotation to a language genetically affiliated to those languages for which they were designed. To have confirmed the extensibility of an already-important annotation standard is itself an important contribution to the resources available for morphosyntactic tagging.

7.1.1.1.1 Possible improvements to the tagset

That said, it would be foolish to claim that the U0 tagset is optimal. Although it represents the best that could be achieved within the scope of this project, there is still room for improvement. In this section I outline some potential improvements which became evident in retrospect towards the end of the project.

One simple improvement would be to alter the tags for punctuation to make them hierarchical[1]. The fact that each punctuation mark is tagged as itself makes it impossible, in the Unirule formalism, to write conditions that refer to all punctuation tags using a wildcard character – something which would, at times, have been quite useful. Ironically, the benefits of the hierarchical tagset were most notable in the only area of the tagset where they were absent.

Another problematic area of the tagset which might possibly be rethought is the classification of nouns into two inflectional categories, marked and unmarked. As

[1] See 2.2.3.

mentioned in 6.2.3.3, loanwords from Arabic and English frequently do not conform to the pattern of unmarked plurals, though they lack the suffixes that would distinguish them as marked nouns. This creates problems for the analyser: the morphological analysis algorithm described in 6.2.3.3 is designed to cope with the standard unmarked and marked classes, not with English or Arabic patterns. A better solution might be a three-way classification, although this would introduce yet more ambiguity into an automatic analysis and make manual analysis that much more difficult. Alternatively, the paradigm classification could be removed entirely, making just a single class of "noun". However, this could also lead to difficulties for the morphological analyser, as it would further complicate the ending-category mappings.

A change in the tagset which would greatly reduce ambiguity in the output of the automatic system, and simplify the task of manual annotation, would be to merge the categories for formally identical classes. For example, it might be possible to use a single tag for all marked feminine adjectives. However, it is questionable to what extent this would be an improvement. Tagging would be made easier, but at the cost of the informative power of the analysis. However, it would be a worthwhile direction of experimentation.

There are doubtless many other alterations to the tagset which might be of benefit. Unfortunately, it was not practical within the limits of this study to follow up any of these. However, it remains a possibility for the future.

7.1.1.2 The tagging system

It is trivially obvious that the non-Urdu-specific elements of the tagging system (i.e. Unilex, the Unirule software and formalism, and the Unitag framework

itself) represent a resource that was not hitherto widely available to the corpus linguistics community – that is, basic tagging software designed to function with two-byte Unicode text. However, the Urdu-specific elements of the system – the Urdutag analysis program, the manual lexicons, and the Urdu rule list – represent a resource of greater, if narrower, utility. With these automatic part-of-speech analysis of Urdu Unicode text is made possible[2].

The performance levels of the Urdu tagging system are another question. As outlined in the previous chapter, the results when running the full system on the training dataset were 99.0% accuracy with an ambiguity of 1.73 tags per word. However, running on the test texts, the same system achieves 90.6% / 2.20 (spoken text) and 88.1% / 2.97 (written text). As was pointed out in 5.6.1, it is very difficult to meaningfully compare the performance rates of different taggers. However, it seems clear that the Urdu tagger described in this thesis does not match up to the mainstream of taggers for languages such as English. Markov model taggers for English regularly score above 95% accuracy with ambiguity 1, for instance. In the next section I will discuss reasons for this relatively low performance; in 7.1.1.2.2 I will discuss possible improvements to and extensions of the tagger.

7.1.1.2.1 Reasons for relatively poor performance in the tagging system

Many factors may well be involved in determining the performance level of the tagger. One known, if trivial, factor is that the benchmark files against which the tagger's output is evaluated are known to contain errors. These files were not

[2] As a result, the tagging system described in this thesis has been used to tag the Urdu part of the EMILLE Corpus (see 2.3).

extensively scrutinised during the processes of lexicon creation and rule writing (which is how most of the errors in the training data were detected). The manual annotator is known to have made consistent errors which the automatic tagger does not. These errors in the benchmark inevitably reduce the accuracy scores when running on the test data.

However, it seems clear that the primary cause of the tagger's poor performance is an inadequate lexicon. Running on its training data, where it benefits from a lexicon containing all the words, it performs well. On the test data, where many tokens are not in the lexicon, accuracy drops by 9-10%, and ambiguity increases drastically. Further evidence that the problem lies in the lexicon can be found in the results of the experiment looking at the lexicon "threshold" in 6.3.3.2. The accuracy of the written text in particular barely declined at all as the threshold rose and the size of the lexicon decreased (see Figs. 6.3, 6.4). This suggests that the common core of Urdu vocabulary which needs to be in the lexicon for the tagger to cope with unseen text has not been captured by deriving a lexicon from the training data. This is also suggested by the size of the lexicon – at the most, circa 3,900 items. As a point of comparison, the English handcrafted tagging lexicons used by the CLAWS tagger (Smith 1997: 141-144) contained 15,000-23,000 items, and some automatically derived lexicons rose to 45,000 items – an order of magnitude greater than the largest Urdu lexicon I could possibly construct. It can therefore be seen that the small size of the lexicon, a result of the scarcity of training data explained in Chapter 4, hamstrings the Urdu tagger from the outset.

One might expect the morphological analysis component of Urdutag to do something to mitigate this, by providing a suitable set of tags for unknown words. However, in practice, this does not seem to be happening. This is probably due to a

combination of two factors. The first is the fact that almost no word terminations in Urdu indicate exclusively a single category or even a small group of categories. Instead, a single ending may indicate a larger number of categories (for instance, *–ē* which may indicate NNMM1O, NNMM1V, NNMM2N, JJM1O, JJM2N, JJM2O, RRJ, VVST1, VVSV1, VVYM1O, VVYM2N, and VVYM2O). There are also very many words which end in the string in question by coincidence, and fall into some other category. It was impossible to capture all of these in the manual "exceptions" lexicon (6.3.2), because of the small amount of manual training data. The second factor is the very large number of unknown words that display no ending, particularly Arabic loanwords.

However, it seems clear that given an appropriate lexicon, the disambiguation rules devised for Urdu do work well. On the training data, they reduce ambiguity from 2.55 to 1.73 tags per word (removing over half the ambiguity in the initial analysis) at a cost of only 1.0% accuracy. It is on the test data, where due to the lexicon the analysis is poor to begin with, that the disambiguation rules cause an unfortunately large number of errors (decreasing accuracy from 89.9% to 88.1% on the written test data, and from 92.5% to 90.6% on the spoken test data).

It would therefore seem that of the resources created during this study, the Urdu lexicon is the weakest and least adequate. Unfortunately, this leads to a comparable inadequacy in the tagger as a whole. However, the software tools that created the lexicon would provide a means to create a far superior lexicon, if only adequate training data were available.

7.1.1.2.2 Possible improvements to the tagging system

The most obvious means of improving the tagger is clearly to use a better lexicon. The most straightforward way to do this would be to tag a large quantity of text (hundreds of thousands or millions of words) using the tagger as it is, and then to manually post-edit this text to achieve full accuracy. This data could then be used as a new training dataset to acquire a better lexicon. This is a process so intensive of native-speaker analyst time that it could not be undertaken within the scope of this project. An alternative approach would be to construct a better lexicon manually, perhaps with the aid of a dictionary. However, this too would require a native speaker, if a monolingual dictionary or intuition alone were used. Using a bilingual dictionary such as Haq (2001) would not help matters because a native speaker's intuition would still be needed to translate the very broad morphosyntactic information given by Haq to the specific information required in a U2 tagging lexicon. In summary, to create a lexicon capable of achieving high accuracy rates on texts that did not form part of the dataset from which the lexicon was derived would be a time-intensive and expensive procedure, no matter how one went about it.

Another enhancement to the system would be the addition of some means of "idiom tagging" to the disambiguation phase of the Urdu tagger. In the system outlined in Chapter 6, some idioms (i.e. consistent phrases which should be consistently tagged) are handled by means of disambiguation rules (e.g. *bī bī sī*, *alsalām 'alaikum*). This is a rather lengthy and roundabout way to do it. Other taggers have profitably employed an independent idiom-tagging module (e.g. CLAWS: see 5.3.2.2, 5.6.3). This is something which might usefully be added to Unitag.

During the process of rule-writing – and therefore when it was too late to

modify it – certain drawbacks in the Unirule formalism that limit its flexibility became apparent. The most serious of these was the inability to create complex conditions. The Unirule formalism (6.2.4.2) only allows a list of conditions to be specified in each rule, all of which must be fulfilled for the action to be triggered. During rule writing, there were many occasions when a rule or group of rules could have been made more perspicuous and economical if complex conditions involving AND, OR and NOT operators, and bracketing conventions as in a programming language, had been possible. An ELSE operator, to specify an action that should be undertaken if a condition is not fulfilled, might also be useful in a revised version of Unirule.

Finally, although I intentionally excluded it from the Unirule formalism in order to avoid complicating the formalism (see 6.2.4.2.2), the experience of rule writing has convinced me that it would be beneficial to allow rules to refer to ranges of "X tokens or more/less" (as is possible in Constraint Grammar).

7.1.2 The duration of the project

The development of the tagger was undertaken on a very tight schedule. Although the entire study stretched over a period of slightly less than three years, the majority of that time period was devoted to necessary preliminary steps, rather than to the construction of the tagger itself. Some of these steps have been described in this thesis, for example devising the tagset and undertaking a phase of manual tagging. Others have been passed over in silence as tangential to the topic of the thesis. For instance, a considerable amount of work on the basic computing that underlies the tagger was necessary – for example, creating basic functions to handle Unicode text in

C. It was also necessary to collect the Urdu text to be tagged, i.e. to construct the Urdu corpus in the first place, which was itself something of a challenge.

The development time for the tagging system itself was therefore quite short – only four months towards the end of the project. This included devising the Unitag and Unirule formalisms, writing the programs described in the previous chapter (although some elements of Verticalise and Unilex were in place earlier on), editing and correcting the training data, creating and optimising the lexicon, and writing the rule list. By comparison, devising the tagset occupied most of the first year of the project, and background investigations into tagging methodologies most of the second. A large part of the third year was invested in the phase of manual tagging. Of course, at any given point in the project, work was undergoing on several aspects simultaneously.

The system described in this thesis may therefore be seen with some justification as a rapidly prototyped tagger. It follows that the creation of this tagging system tells us not only that morphosyntactic tagging of a new language can be done; it also tells us that it can be done *quickly*. This knowledge is in itself a valuable result of this study.

It should be noted that this aspect of my research is fairly timely. Recent investigations sponsored by DARPA (the American government's Defence Advanced Research Projects Agency) have tested the feasibility of developing natural language processing applications and resources for a "new" language (in this case Hindi) in a very short period of time, one month[3]. In such a context speed of development is

[3] As this "Surprise Language Project" is very recent work (June/July 2003), I am unaware of any publications discussing it. However, a report of some of its results is available on the World Wide Web at http://www.usc.edu/isinews/stories/98.html .

clearly of great importance. This study has gone some way towards demonstrating that such speed is achievable in the field of part-of-speech tagging.

7.1.3 Discoveries concerning the structure of Urdu

The primary aim of this thesis was clearly not to make new discoveries about the structure of Urdu. However, in the course of the study, some findings in this area have been made. For instance, the process of rule-writing brought to light some details about the structure of Urdu (see section 6.4.4).

The overall process of designing and analysing the tagger has brought some other factors to light. In particular, the high ambiguity and low accuracy rates achieved on texts which were not used to derive the lexicon suggests that the category of a large proportion of Urdu words is not easily predictable from their form. This is a fairly counterintuitive result, given the comparatively large number of inflections that may be found in Urdu for gender, number etc. In turn, this suggests that large numbers of words in common use in Urdu are Persian, Arabic or English loanwords, since these words may typically deviate from the inflectional paradigms of Urdu. The actual extent of loanwords in Urdu as commonly used was an unknown factor in the development of the tagger (as discussed in 6.2.3.3). At the end of the study, it seems likely that their extent is significant enough that any analysis tool must be designed to handle them very frequently. This is an important discovery to have made.

However, the major discovery that this study has made, as far as the structure of Urdu is concerned, is that the model of the grammar provided by Schmidt (1999), is an adequate model for practical applications in Urdu language engineering (with the minor drawbacks discussed in 4.2.1.7). Knowing the applicability of Schmidt's

grammar to the field will allow future researchers in Urdu language engineering to make use of the model without uncertainty as to its suitability. This is clearly a notable advance.

7.2 Possible future research

An obvious extension to this study would be to compare the Urdu tagger described here with a pre-existing tagger retrained for Urdu. This has not been attempted within this study because of many difficulties that have been outlined earlier in this thesis, for example, the difficulty in converting text back and forth between Unicode and ASCII, or in training a tagger in the face of the paucity of training data, and of course the difficulty of a meaningful comparison between two taggers in the first place

While the creation of an Urdu tagger represents an important annotation tool for use with Urdu corpora, there are other tools which could beneficially be developed in this area. A parser for Urdu would be an obvious next step. Not only could a parser capitalise on the analysis performed by the tagger, it might also be able improve the quality of the tagger's output (see 5.4.1 for a discussion of the use of a parser in tagging).

Another relatively simple "next step" would be to extend the work done on the Urdu tagger to Urdu's nearly-identical sister language, Hindi. A new lexicon would be needed, and a new analyser program, since the vocabulary and the writing systems are the areas of Urdu and Hindi that vary most from one another (see 1.1.4). Changes might also be needed to the way tokenisation is handled. However, the major part of the software, and in particular the rules, could probably be applied to Hindi with only

minor modifications. The experience with Urdu might also be a very good starting point for attempting morphosyntactic annotation in other Indo-Aryan languages, which have the same prior requirements as Urdu in terms of Unicode-compliant software.

7.3 An overall summary of the thesis

I will now proceed to summarise this thesis, looking at the progress made in each chapter.

In Chapter 1, I provided a background introduction to this study, looking at some basic information on Urdu and part-of-speech tagging in corpus linguistics, and explaining the interest creating a tagger for Urdu within the context of the EMILLE project.

I then proceeded to the creation of a tagset. Firstly, in Chapter 2, I looked at the preparatory steps necessary for a tagset definition, addressing three main aims. The first of these was a discussion of the history of tagset design, so that the Urdu tagset might be based on established best practice. Secondly, I established and justified a set of design principles for the tagset, including, most importantly, adherence to the EAGLES guidelines, a major multilingual standard. Thirdly, I selected the grammar of Schmidt (1999) as a model of Urdu grammar for use in the creation of the tagset, and justified this decision. Chapter 3 used this basis to move onto the actual definition of the tagset. As well as creating the U0 tagset, I was concerned to substantiate my claim that the EAGLES guidelines were a suitable basis for the Urdu tagset. Since they were not written with Urdu in mind, using them in this way constitutes an extension of their usefulness; I have demonstrated their

extensibility in this fashion (see the discussion in 3.15).

Chapter 4 describes a stage of manual tagging which was undertaken using the now-established tagset. My aim here was to justify such a stage within the context of a project aimed at automated tagging. One of its benefits was that I was able to confirm the applicability of Schmidt's (1999) grammar to this kind of application. Other benefits included the possibility of identifying points at which it was necessary to diverge from the model presented by Schmidt, and the opportunity to identify categorisation difficulties, which I also outlined in this chapter. I also argued for claims regarding the necessity of tagging guidelines, whose creation fulfilled a secondary aim of the thesis, and the need for the use of the U1 and U2 subtagsets, which I described.

In Chapter 5 I moved away from the topic of Urdu, to consider work previous conducted in the area of part-of-speech disambiguation (disambiguation being the part of the tagging process where methods are most diverse). I reviewed four broad groups of tagging methodologies: rule-based approaches, probabilistic approaches, approaches using corpus-derived rules, and machine-learning approaches. I then justified my choice of a methodology for this project. While difficulties in comparing different taggers made it impossible to choose a methodology based on the results they have achieved, practical restrictions on this project – in particular the small amount of training data available – made it necessary to use a hand-crafted rule-based approach to disambiguation.

Chapter 6 presented the implementation of the automatic tagger. I gave an overview of the main software components that I wrote for the Urdu tagger – particularly the Unitag framework, the Urdutag analyser, the Unilex lexicon creation software, and the Unirule disambiguator and its accompanying rule formalism. I also

outlined the creation of the lexicons and the rule list. I described a number of experiments conducted with the aim of optimising the lexicon, the morphological analysis of unknown words, and the performance of the rule-list in disambiguation. The net result was the tagging system whose final performance has been assessed above.

Finally, in this conclusion, I have summarised the results of the study in terms of the resources developed and the discoveries made. I have also noted potential improvements to the resources that are apparent in retrospect, and noted some possible avenues of future research in this and related fields.

7.4 Concluding remarks

If this thesis has had a single aim, it has been the demonstration that automated part-of-speech tagging of Urdu text saved as Unicode is possible using pre-existing knowledge, techniques and standards – the knowledge of Urdu grammar expressed in Schmidt (1999), the rule-based disambiguation methodology, the EAGLES guidelines, and so on. The fact that it has been possible to produce a working system quickly and with relatively minimal native speaker input is proof that this aim has been fulfilled. This has been done by a synthesis of two strands of prior linguistic research to which I have referred in turn throughout this thesis: into Urdu and into techniques of corpus analysis. This approach has allowed a tagger to be created for a language which has not previously been addressed in this field of study, in a relatively short period of time, as discussed above.

Nevertheless, as has been pointed out above, there are flaws and room for improvement in the tagger. While comparison between different tagging systems is

very difficult, as I have pointed out several times, it is nonetheless very clear that the Urdu tagger described here does not approach the levels of accuracy and ambiguity that have been achieved for languages like English. It is not to be expected that a single small-scale project such as this could match the result of at least two decades' intensive research. But it *has* been possible to create a working system capable of producing output that is linguistically useful. While this study cannot represent the last word in terms of the computational analysis of morphosyntactic categories in Urdu, it has certainly made significant progress in the development of this technology.