ON THE SCALABILITY OF THE ANSWER EXTRACTION SYSTEM "EXTRANS"

Diego Mollá   Michael Hess

There have been many attempts in the history of Information Retrieval (IR) to add some linguistic capabilites to standard IR systems in order to improve their performance (mainly, their precision).[1] These attempts have not been very successful so far, at least not in the standard IR settings (cf. [7]). The two main reasons are the (related but not identical) problems of *data volume* and of *scalability*. First, the volume of data typically processed by IR systems is so large that the use of more than a few isolated linguistic components seemed out of the question, and linguistic components do not work well in isolation. Second, NLP systems that work reasonably well in small scale laboratory contexts will often not scale up to real world domains like those for which IR is standardly used. Both of these points seem to all but rule out the use of full-fledged NLP methods in standard text retrieval applications.

For some specific applications, however, high recall and precision are even more crucial than in IR yet the volumes of data to process are much smaller. These applications include interfaces to machine-readable technical manuals, on-line help systems for complex software (such as operating systems), help desk systems in large organisations, and public inquiry systems accessible over the Internet. In all these applications the document collections to be accessed are just a few hundred megabytes in size at most. The users of these applications do not want a set of complete documents, each one possibly dozens of pages long, as in standard IR. What they want is a few highly specific answers to their highly specific queries. In other words, in such applications the user needs a system that locates those exact *phrases* in the documents which contain the explicit answers to their queries. This is what an *answer extraction system* is supposed to do, and it will require the use of linguistic knowledge if it is to succeed. Note that answer extraction systems are not meant to *infer* answers from implicit information contained in the documents (as is the idea of full-fledged text understanding systems). All they should do is retrieve phrase-sized passages of text containing an *explicit* answer to a query, if there is one. This is the task called, a bit confusingly, "question answering" in TREC-8 ([8]).

We are currently developing such an answer extraction system for the online Unix manpages. The system, ExtrAns, uses NLP as its core technology to achieve the needed performance in terms of very high precision and recall for highly specific queries [5, 1].

[1]This research is funded by the Swiss National Science Foundation, project nr. 12-53704.98

The overall structure of ExtrAns is shown in figure 1. At indexing time, i.e. *offline*, the manpages are subjected to a full syntactic analysis, integrating the system developed by [6]. We then weed out obvious wrong readings of the input sentences by using a set of specific rules. The words are then converted to their base form by using the lemmatiser included in [3]. More difficult ambiguous readings are deleted by adapting and enhancing [2], a corpus-based disambiguator. Next, intra-sentential pronominal references are resolved, following the algorithm suggested by [4]. Finally, logical forms are derived, transformed into Horn clauses, and stored in a database.

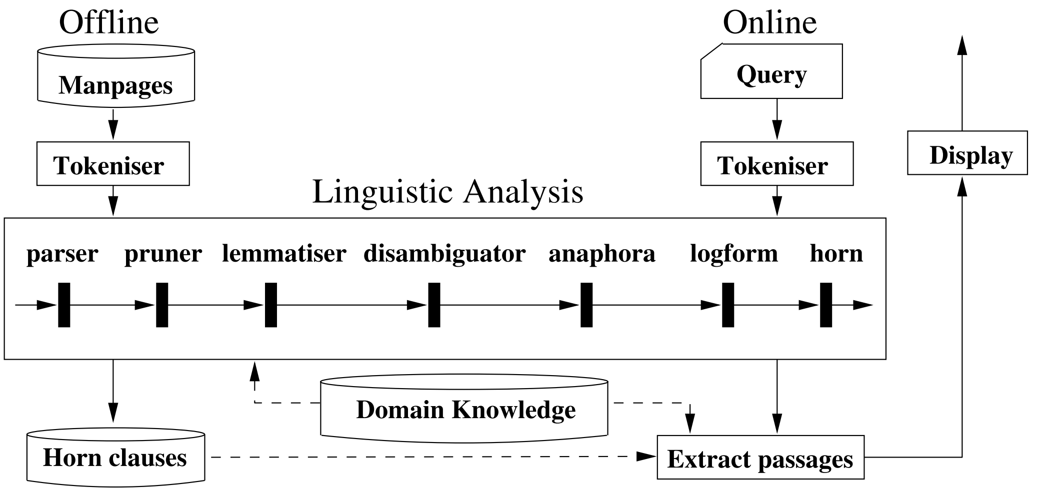

Figure 1. Structure of ExtrAns

The user can then freely query the system by asking questions in plain English. The logical form of the query is computed, *online*, in the same way as described above, and the system tries to prove the query over the database. If successful, the corresponding sentences are retrieved and displayed, with those phrases that explicitly answer the user query highlighted according to their relevance to the query, where relevance is a factor of (un)ambiguity [5]. Figure 2 shows the logical forms of a query and an answer (see [5] for more details about logical forms), and figure 3 shows a screen shot of the output of ExtrAns for the same query.

which command erases files?
```
object(s_command,A,B), evt(s_remove,C,[B,D]), object(s_file,E,D)
```
rm removes one or more files
```
holds(v_e2), object(rm,v_o_a1,v_x1), object(s_command,v_o_a2,v_x1),
evt(s_remove,v_e2,[v_x1,v_x6]), object(s_file,v_o_a3,v_x6)
```

Figure 2. The logical form of a user query and an answer (simplified to ease readability)

In order to test scalability, we started with a small subset of 30 Unix manpages as development set and extended the document basis, in a second step, to a test set of over 500 manpages. We tuned the

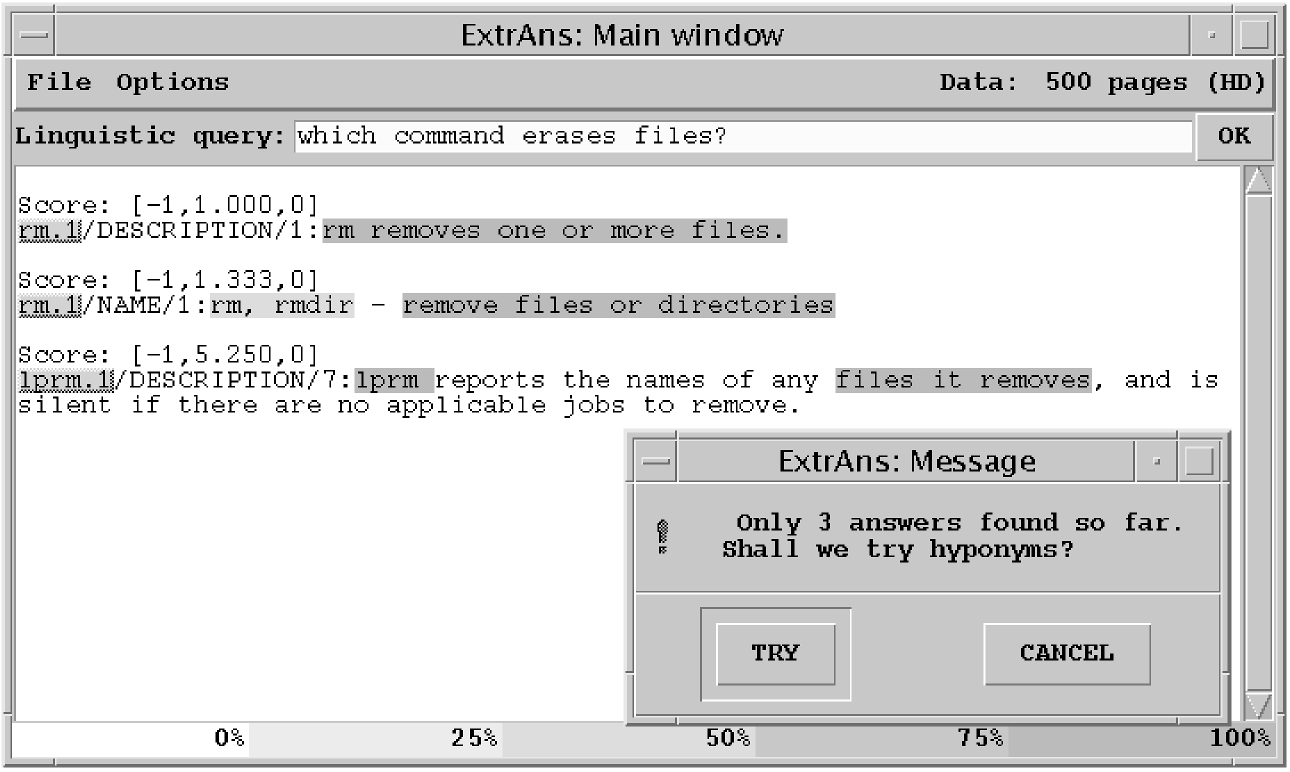

Figure 3. A screen shot of ExtrAns (monochrome rendering)

system so as to be able to cope with the larger volumes of data but did not change any of the linguistic components (we only extended the lexicon to increase the accuracy of the parses, although the parser used [6] does handle unknown words). A brief description of the modifications follows:

1. The set of Horn clauses was stored in an external database, since a larger size of the data would not fit in the RAM memory. An obvious consequence of this decision is that the speed of retrieval would degrade, but we solved this in the second set of modifications:

2. The database of Horn clauses was divided into a set of databases, one for each manpage, and a pre-selection step was added so that the query is run only over those manpages that contain all of the terms used in the logical form of the query. In this fashion, only those manpages that are likely to contain the answer to the question are examined.

We found that the system retrieved the same sentences as before and that response times, after performing the modifications, were shorter, even if the number of manpages treated was considerably larger. In table 1 we can also see that the relative increase of response time is well lower than the ratio of the sizes of the data of both sets of manpages ($272\text{Kbytes}/20\text{Kbytes} = 13.6$). The original system thus turned out to be perfectly scalable, contrary to what is normally assumed to be the case for NLP-based retrieval systems.

Sentence	30 manpages **Internal** (a)	**External** (b)	500 manpages **External** (c)	(c) / (b)
which command copies files?	2980	584	3206	5.48
how can I create a directory?	7412	1270	3374	2.65
which command removes directories?	4632	416	1080	2.59
how can a file be removed?	5110	936	4584	4.89
can I remove some columns from a text file?	5750	316	384	1.21
what is ipcrm?	258	194	246	1.26
which command erases files?	7430	794	3986	5.02

Table 1. Response time in *ms.* from two sets of manpages, on a 167-MHz UltraSparc machine. The set of 30 manpages was stored in an internal (a) and an external (b) database. The set of 500 manpages was stored in an external database (c).

The idea of preselecting manpages is connected with another possibility that we are considering. We could include a standard Information Retrieval (IR) module specifically tuned-up to give results with high recall. The IR module would provide ExtrAns with a reduced set of data, and ExtrAns would use its linguistically-aware techniques to further reduce the amount of data, so that eventually the user would get the wanted answers with a high index of recall and precision. By combining standard IR techniques with a system such as ExtrAns it would be possible to find answers to queries over data in the size scale of gigabytes.

By combining available linguistic resources and implementing only a few modules from scratch, we have been able to put together a system with full linguistic analysis in a relatively short period of time (4 man-years). Table 1 shows that the increase in time is not a big issue when scaling the system up from 30 to 500 documents. The main result of our experiment is thus that the use of existing NLP techniques allows us to implement answer extraction systems whose performance is perfectly acceptable, even in their laboratory versions, and that such systems do scale up to practical dimensions.

[1] BERRI, J., MOLLÁ, D., HESS, M., Extraction Automatique de Réponses: implémentation du système ExtrAns, in: Proceedings of the fifth conference TALN 1998 (Traitement Automatique des Langues Naturelles), Paris 1998.

[2] BRILL, E., RESNIK, P., A rule-based approach to prepositional phrase attachment disambiguation, in: Proceedings of the 15th International Conference on Computational Linguistics (COLING '94), vol. 2, Kyoto, Japan 1994.

[3] GAIZAUSKAS, R., CUNNINGHAM, H., WILKS, Y., RODGERS, P., HUMPHREYS, K., GATE: an environment to support research and development in natural language engineering, in: Proceedings of the 8th IEEE International Conference on Tools with Artificial Intelligence, Toulouse, France 1996.

[4] LAPPIN, S., LEASS, H. J., An Algorithm for Pronominal Anaphora Resolution, Computational Linguistics, vol. 20 (1994).

[5] MOLLÁ, D., BERRI, J., HESS, M., A Real World Implementation of Answer Extraction, in: Proc. of the 9th International Conference and Workshop on Database and Expert Systems. Workshop "Natural Language and Information Systems" (NLIS'98), Vienna 1998.

[6] SLEATOR, D. D., TEMPERLEY, D., Parsing English with a Link Grammar, Tech. rep., Carnegie Mellon University, School of Computer Science 1993. Presented at the Third International Workshop on Parsing Technologies.

[7] STRZALKOWSKI, T., GUTHRIE, L., KARLGREN, J., LEISTENSNIDER, J., LIN, F., PEREZ-CARBALLO, J., STRASZHEIM, T., WANG, J., WILDING, J., Natural Language Information Retrieval: TREC-5 Report, in: E. M. Voorhees and D. K. Harman (eds.), The Fifth Text REtrieval Conference (TREC-5), no. 500-238 in NIST Special Publication, Gaithersburg 1997.

[8] TREC-8, Call for Participation Text REtrieval Conference 1999 (TREC-8), http://trec.nist.gov/cfp.html 1999.