CSE 705: Seminar Report on

Towards Best Effort Information Extraction: iFlex

Akanksha Dayal (adayal@buffalo.edu)

Overview:

The iFlex system presented in this paper puts forward a very different approach for Information extraction by relaxing the preciseness requirement that the traditional IE systems bear. The system presented in this paper enables the user to write an initial approximate Information extraction (IE) program 'P' which is given to an approximate IE processor to get the approximate results. The system takes an iterative approach, i.e. the user 'u' can repeatedly refine the program 'P'. This approach alleviates the drawbacks of most of the existing IE systems such as long debug loop and incompetence for time sensitive applications. This also saves time when an approximation to a result set is suffices the user's needs.

As it goes, the user 'U' uses a declarative language and writes and initial approximate IE program 'P' using an approximate Query Processor that may fetch an approximate result. 'U' then examines that result and figures out if further refinement is required. In case a further requirement is desired, the user 'U' can either manually modify 'P' or 'U' can employ the *Next Effort Assistant* 'A' to suggest what to focus on for example, look for the prices in bold. This way 'U' knows when the program is under-defined or over-defined by looking at the result set. More time spent writing program/iterating leads to high precision results.

The language chosen to write approximate programs is Alog that heavily derives from Xlog - a datalog variant used for writing declarative IE programs. The user can go on adding domain constraints, that take the form f(a) = v : feature f of any text span that is a value for attribute a must take value v to the rules specified while writing P. IFlex provides inbuilt function like from(x,y) that ensures writing safe description rules.

Approximation is achieved in the system by providing the supporting the two annotations - existence of a tuple and value of an attribute. The approximate results are represented using compact tables since it already provides support for the two assignment types – exact and contain. The program is compiled using an approximate query processor.

The system is optimized by incorporating Reuse and Subset evaluation. Reuse refers to reusing the intermediate results generated in the previous iteration to avoid the re-computation thus saving time and efforts. Subset evaluation refers to executing the plan p over only a subset of the input documents, thus dramatically reducing execution time.

Detail Comments:

Strengths:

- The best effort information extraction approach presented in this paper operates faster as compared to the traditional approaches towards information extraction.
- The system saves time and effort in scenarios when an approximation of the results is more than sufficient. This is due to the relaxation in the preciseness of results generated.
- The language specified is declarative i.e. lacks a procedural code implementation. As a result this approach is well suited for time sensitive applications. The results received by this approach are reasonably better than those fetched by the traditional systems.
- This system enables user to execute partially specified code preventing long debug loops.

Weaknesses:

- The user 'U' in this system needs to have an understanding of the extended datalog variant language *Alog* in order to write the p-predicate and p-functions.
- Sometimes, the user might still not be satisfied with the results even when the system converges. The user has then no option but to write a p-function for further refinement and provide an implementation of it in a procedural language like perl or java.
- The system does not apply to scenarios that ask for the precise IE results.
- The approach does not address non text specific requirements of data extraction.

Technical Depth:

The paper is technically very sound as it covers every aspect of the approach in detail. It talks about the challenges faced while realization of the system and the methods/strategies adopted to counteract them. For example the use of superset evaluation (approximate approximation) to counter the problem – 'approximation result representation model not being complete under relational operators'. The paper also compares the options available for realization of a step and then picks up the one that answers the system needs best (like for representing approximate results, it first compares the possible ways of approximate representation in results i.e. a-tables and compact tables and then based on the desirable selects the compact table).

Comparison with related work:

Works by Benjelloun, L. Antova, T. Jansen, N. Dalvi and D. Suciu to name a few, have focused on the representation and processing of approximate or uncertain data in general , however, they have not considered the text-specific challenges that are addresses in the iFlex approach. The approach studies the representation of the large number of possible extracted values a best-effort IE program may produce. Also, the interactive query processing has been studied in the control project by Hellerste for data analysis in relational settings unlike as in IE over text as is presented in this paper.

Discussions:

Audience questioned about the use of ranking measure if any for the presentation of the results to the user. Also we discussed about the compact table format being the output to the user. Some of the audience was inquisitive about the annotation operator used to convert set of possible relations by applying the rules in P. Audience was also interested and pointed out the use of library functions in the iFlex.