Evaluation of Intelligent Information Retrieval Tools for Unstructured Police Case Reports

Nishant Kumar, Jan De Beer, Jan Vanthienen and Marie-Francine Moens
Katholieke Universiteit Leuven, Belgium

***Abstract*— The Belgian Federal Police (BFP) already using an array of advanced and task-specific tools for collection, dispersal and exploitation of highly structured information, is now shifting attention towards intelligent information retrieval technologies for exploiting vast amounts of unstructured information sources (police case reports), from which further knowledge could be obtained for analysis purposes. The INFO-NS research project, is aimed to provide an objective study of the applicability of mining and decision support tools for the BFP. More specifically, it is studied whether information retrieval, extraction and classification tools might leverage intelligence and decision support while exploiting the information that is contained in vast amounts of free text material and linking it with any coexisting, structured data sets that are currently in use. This article provides an insight into the requirements of the BFP and police in general with regard to a like decision support system, gives an overview of our evaluation approach and presents one aspect of evaluation criteria aimed at testing the functional conformity of such tools against a set of functional requirements.**

***Index Terms*— information retrieval, text mining,law enforcement, Knowledge representation, evaluation techniques, multi-criteria evaluation**

I. INTRODUCTION

THE research project INFO-NS is an initiative of the Belgian Federal Police (BFP), commissioned by AGORA as part of the Belgian Science Policy,[1] and carried out by the Katholieke Universiteit Leuven. It is part of the larger DOCMAN project of the BFP, which is aimed at storing the pv's (police case reports) and their base metadata in a central database and to make them accessible through the police's intranet.

Although the information contained in pv's is painstakingly structured into a relational and readily exploitable data format, the structuring process has its drawbacks. In particular, information that is often ambiguous and vague in nature is converted into a crisp and fragmented representation, with subtle yet important nuances being lost in the process, and the big picture becoming more difficult to grasp. Moreover, structuring millions of documents on a yearly basis is a massive undertaking, with even greater amounts of documents left unstructured. For these and other reasons, the use of search and text mining tools might prove a viable solution. These tools enable to index and classify unstructured information and to annotate them with additional metadata. The better this information can be indexed, classified and labelled, the more accessible and useful this information becomes for the users. These tools can support the retrieval, querying (searching), management, structuring, visualisation and extraction of relevant information from (electronically available) textual and multi-media documents, within the different services of the BFP. In addition, they allow searching complementary textual sources such as analysis reports, investigation notices, news stories, and webpages.

Many of the text mining and search tools available today are extremely fast, powerful, and easy to use, which makes them appropriate for live environments, such as task forces or operational planning sessions. As the number of text mining software vendors increases, it has become more challenging to assess which of these tools are most effective for a given application. Such judgment is particularly useful for both purchasers of text mining tools given the high investment (money and time) required in becoming proficient in their use, and developers who aim at producing better quality text mining products. The main aim of the INFO-NS project is therefore to evaluate a number of text mining tools with regard to their applicability, their performance and their capabilities of integration within the current BFP infrastructure. As a special concern, the multilingual and crosslingual support for the three official languages in Belgium (Dutch, French, and German) constitutes an important consideration.

An important research question we want to answer in the project is how to objectively and reliably evaluate the tools so that we have an answer to the following questions.

- To what extent does each of the preselected tools (a short listing from market) answer the functional needs of a police administrator, investigator, operational and strategic analyst?
- To what extent does the tool perform at quality levels like capability, accuracy, flexibility etc.?
- How efficiently are system resources (memory, disk space, network bandwidth,…) utilised, considering large document collections, and how well does the tool meet the other criteria like user-friendliness, the system/user interaction, the quality of documentation, etc.?

N. Kumar and J. Vanthienen are with the Leuven Institute of Research in Information Systems (LIRIS) group at the K.U.Leuven.
E-mail: {nishant.kumar,jan.vanthienen}@econ.kuleuven.ac.be
J. De Beer and M. F. Moens are with the Legal Informatics and Information Retrieval (LIIR) group of the Interdisciplinary Centre for Law and ICT at the K.U.Leuven.
E-mail: {jan.debeer,marie-france.moens}@law.kuleuven.ac.be

[1]Visit AGORA at http://www.belspo.be/belspo/fedra/prog.asp?l=en&COD=AG

II. Project Description

A. Outline

Figure 1 gives an overview of the INFO-NS project. Before implementing a decision support system for the police to help them in investigating criminal activities, some vital questions need to be answered with regard to the profile of the users and their requirements. User profiles will be presented further in the text, where we work out the functional evaluation of tools. Concerning requirements, we discern information needs, functional needs, and technical needs.

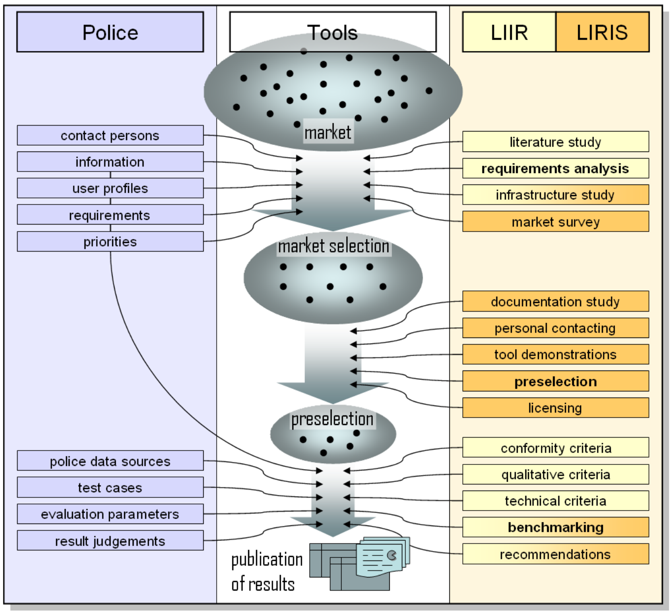

Fig. 1. Schematic view of the outline of project INFO-NS, indicating each partner's share as the project progresses through successive stages (downward orientation on the scheme).

1) Information needs: Under the information needs we consider the unstructured sources of information for which exploitation is desirable, taking into account the objectives and modalities of the DOCMAN project, which at present is under execution. A potential number of different types (internal and external) of information sources have been considered, including case reports, multimedia documents (differently than text), police internet and intranet sites, and the library of the their documentation centre.

2) Functional needs: Under the functional needs we consider the desired functionalities for the exploitation of the sources of information identified through our study. An overview is presented further in the text. The desirability of these functionalities for each of the user group has been identified and a priority list of functionalities has been prepared.

3) Technical needs: Under the technical requirements we consider the efficient use and allocation of system resources and various usability aspects that need to be considered for an eventual, large-scale deployment of the tools in a live environment.

B. Evaluation approach

For the purpose of evaluation, we identified three major groups of evaluation criteria, capturing the applicability, the competence, and the practicality of the tools under study.

- **Applicability** The extent to which each of the preselected tools (an initial market selection) answers the identified functional needs of the various user profiles.
- **Competence** The extent to which each of the tools performs at quality measures like capability, accuracy, flexibility, scalability, etc. For this purpose, task-specific evaluation procedures and criteria are devised.
- **Practicality** Includes performance, as the extent to which system resources (memory, disk space, network band width,...) are efficiently utilised, considering extensive document collections and a large potential number of concurrent users, next to various, more subjective criteria, such as user-friendliness, user-system interaction, the quality of documentation, etc.

For the remaining of this paper, we restrict ourselves to the application of the first of these groups, coined conformity evaluation.[2]

III. Conformity Evaluation

In this section we present our methodology for the evaluation of tools solely on the basis of their support (provision) regarding any functional needs and associated prioritities for a number of distinct user profiles that are identified in the early stages of the project – the requirements analysis phase. The methodology is sufficiently generic, so that it be readily adoptable in other projects, and is quite broad in scope, so as to be readily portable to other situations in which some sort of multi-criteria evaluation or analysis is to be performed.[3]

After we present our methodology, we illustrate how we used this in the context of our project to pursue part of its objective. Given space and confidentiality constraints, we will however not go into too much detail. We end this section with a discussion of our proposed evaluation model.

A. Methodology

Our evaluation model assumes the following information is available.

- A set of tools to be evaluated $\mathcal{T} = \{T_i\}_{i=1}^{t}$.
- A set of relevant functionalities $\mathcal{F} = \{F_i\}_{i=1}^{f}$ with fixed semantics and identifying labels.
- A hierarchy $\mathcal{H}$ defined over the functionalities in $\mathcal{F}$ according to the inclusion relation $\supset$ (read: subsumes); $\mathcal{H} = \{(i,j) \mid F_i \supset F_j \wedge \neg\exists k \neq i,j : F_i \supset F_k \supset F_j\}$. Although not mandatory for our evaluation model, $\mathcal{H}$ puts an order upon a potentially large set $\mathcal{F}$ through the

[2] Publication of the qualitative and technical evaluation procedures is provisional, yet information may be requested from the authors.

[3] We refer to the application of these and similar techniques in police domain for e.g. the prioritization of criminal investigations and the assessment of threats based on offender (group) profiles or environmental conditions.

identification of atomic (indivisible) functionalities and their grouping to more general functionalities. As will become clearer further in this text, $\mathcal{H}$ allows us to proceed in a more methodical and systematic manner.

- For each tool T_i a *support tree* ST_i. This concept is worked out in definition 1 (see below).
- A set of *use cases* $\mathcal{U} = \{U_i\}_{i=1}^{u}$. Formally, every use case represents a logical grouping of related functionalities $U_i = \{F_{u_{i,j}}\}_{j=1}^{u_i}$.
  In practice, a use case represents some particular task which comprises several functional components, in turn consisting out of logically related functionalities. Common components pertain to data preprocessing, the support for accomplishing the task, visualisation and interaction, import-export capabilities, etc.
  The provision of multiple use cases allows the coverage of as many of the functionalities in $\mathcal{F}$ with a selection of any number of tools, given the faint likelihood of having one supertool; a tool that supports most tasks for everyone the best.
- A set of user profiles $\mathcal{P} = \{P_i\}_{i=1}^{p}$ with identified priorities regarding each functionality in $\mathcal{F}$.
- For each use case U_i and user profile P_j a *requirements tree* $RT_{i,j}$. This concept is worked out in definition 2.

Definition 1 (support tree): The support tree of tool T_i, noted ST_i, is a tree structure corresponding $\mathcal{H}$, wherein the node representing F_j carries as attributes the label of F_j for identification, as well as an indication of the degree to which the tool supports F_j.

Definition 2 (requirements tree): The requirements tree of use case U_i for user profile P_j, noted $RT_{i,j}$, is a tree structure corresponding $\mathcal{H}$ restrained to $\{F_{u_{i,k}}\}_{k=1}^{u_i}$. In this structure, the node representing $F_{u_{i,k}}$ carries as attributes the label of $F_{u_{i,k}}$ for identification, as well as an indication of the degree to which $F_{u_{i,k}}$ is desired by users of profile P_j.

Given this information, we now aim to evaluate how well each tool conforms to every use case in $\mathcal{U}$, and this for every user profile in $\mathcal{P}$ individually. As every combination of use case and user profile is reflected in a unique requirements tree, we thus want to compute the conformity between every tool and requirements tree. For this, we define an abstract operator τ that evaluates the "degree of support" of definition 1 with respect to the "degree of desire" of definition 2, given a particular support tree ST and a requirements tree RT.[4]

$$\tau : ST \times RT \rightarrow \mathbb{R}$$

The repeated operation of τ for each tool on all requirement trees then produces an array of conformity scores, which can optionally be combined (through weighing e.g.) to global scores, or used to filter away dominated tools. This latter option can be achieved by retaining only those tools in $\mathcal{T}$ for which there is at least one requirements tree for which they give the best result (among the other tools in $\mathcal{T}$). The selection of non-dominated tools is given by the following formula.

$$\bigcup_{j,k} \{T_i \mid \tau(ST_i, RT_{j,k}) = max_r\ \tau(ST_r, RT_{j,k})\} \quad (1)$$

In the formula, the union is taken of all best tools for every requirements tree.

[4]In the workout, we show how we defined the operator τ.

B. Workout

1) User profiles: In association with the Belgian police, we first identified four user profiles for the tools being sought after. These profiles are quite general in nature and are equally found in other police organisations, even though they may go by different names.

Administrator	Collects, manages, structures, and sometimes already relates facts described in official documents (case reports e.g.), dispatching the gathered or derived information to other services upon request or as part of the information flow.
Investigator	Conducts criminal investigations. Her task is to compile a comprehensive report (a legal case file) describing all acts and elements part of the investigation, which will be the main source of evidence used by judicial authorities for prosecution.
Operational analyst	Examines, supports and assists criminal investigations, especially more complex ones. New hypotheses, alternatives, links, contextualisations, schematisations, etc. can be suggested or provided.
Strategic analyst	Analyses safety problems; their tendencies, trends, patterns, processes, novelties, etc. Such analyses serve as the basis for strategic (long-term) decision making, pinpointing the main security problems and giving insights into their nature and characteristics. This allows allocating limited police resources for top efficacy.

2) Functionalities and priorities: We compiled an extensive list of functional requirements, partly technical requirements of a more prerequisite nature that we as technical researchers were able to identify ourselves, and partly functional needs of the user group, which we gathered through questionnaires, meetings and work sessions held throughout the different police departments. A topical, high-level overview follows.

- Tool Configuration
  – Document content indexing process
  – Security and access control
  – Support for multiple languages and document formats [5]

[5]A prerequisite is the support for the three official languages in Belgium, namely Dutch, French, and German, along with English for open sources. Given the emerging threat of terrorism and the organised crime wave coming from the East, interest in Arabic and Asiatic languages is growing.

  - Inclusion of metadata
  - Automatic clustering or classification
- Search & Retrieve
  - Metadata search: document id, url, title, type, language, origin,...
  - Free text search: crosslingual, fuzzy, conceptual search,...
  - Entity search: crosslingual, phonetic, morphological search,...
  - Similarity search: crosslingual search-by-example
  - Taxonomy search: category or cluster selection
  - Multi-modal search
  - Monitoring: automated signaling of relevant, new or updated information, e.g. through user profiling and proactive search agents
- User-System interaction
  - Assisted formulation of search queries
  - Filtering of search results through successive formulation of queries
  - Relevance feedback and query refinement
  - Repeated search and search history
  - Visualisation, exportation, manipulation, and browsing of search results
  - Automated clustering or classification of the search result
- Qualitative Analysis
  - Discovery of relations between terms, concepts, entities, or any combination
  - Assisted annotation of documents, also known as text coding
  - Support for creation of graphical schemes
  - Automated recognition and classification of entities
  - Visualisation and exportation of analysis results

Functionalities were hierarchically ordered and presented in clear language to police officers of the identified user profiles. By having them score the functionalities to their active needs, we were able to associate real-valued priority values to $\mathcal{F}$ for each profile.

3) Use cases and requirement trees: Out of $\mathcal{F}$ we were able to distinguish ten distinct use cases. As an example, consider the use case “free text search”. As all others, this use case is made up of several functional components, including tool configuration, document indexing, text search, and various interaction functions. Given the hierarchical ordering of our functionalities we set up the corresponding requirements tree, which is depicted in Fig. 2.

4) Tools and support trees: For each of the tools considered for evaluation, we will construct their corresponding support tree.[6] The implementation is done through the specification of *support values* for each of the functionalities in $\mathcal{F}$. Concrete, the support value of tool T_i for F_j, noted $\sigma(T_i, F_j)$, is a real number in unit interval giving expression to the “degree of support” of definition 1. A value of 0 indicates no support, 1 indicates full support, and partial support might be mapped along the continuum.[7]

$$\sigma : \mathcal{T} \times \mathcal{F} \rightarrow [0, 1]$$

5) Conformity matching: In order to match a requirements tree with the support tree of a tool, we implement the matching operator τ through the specification of *objective functions* at every single node in the requirements tree. These functions take as arguments the support values of the tool, the requirement priorities of a user profile, and some extra, profile-independent parameters. Each objective function produces as a result a real-valued conformity score with respect to the functionality associated to the node having the function attached. Through repeated and systematic evaluation of these functions - starting at the leaf nodes, tracing intermediate nodes, and ending at the root node - one obtains a global conformity score for each tool on every use case and for every user profile.

In addition, next to the detailed intermediate results, which can give useful insight as to why and at which points some tools fail, we build two clear and concise contracted tables which we can easily derive through priority composition. One table gives the conformity of each tool for each use case (combined over all profiles), whereas the other table gives the conformity of each tool for every user profile (combined over all use cases).

C. Discussion

1) Considerations: To safeguard the proper application of the proposed evaluation model with a sound interpretation and use of the produced results, a few conditions and remarks should be made.

First of all, support values should be obtained by confident means so as to resemble the tools’ true support, otherwise results are deemed to be meaningless and therefore useless. Through own experience, we found that software vendors have a tendency to badge their products as being extremely versatile and applicable to the specific task at hand.[8] As a researcher, one should therefore strive to establish these values through objective and motivated means, possibly skimming any documentation that describes the tools’ capabilities and features, attending demo presentations, installing evaluation versions, looking for related studies conducted by trustworthy third-parties, through personal use or prior knowledge, etc.

Second, it is the nested objective functions which serve to produce the absolute conformity scoring values. As these functions capture the very semantics of the evaluation (matching) taking place, they should be deviced with great care and precision. Judicious use of mathematical operators (additive, multiplicative, fuzzy logical,...) and overall consistency in design are primal points of attention.

[6]To prevent any market influence and to safeguard the confidentiality of our research, we choose not to make the tools publicly known, at least not at this stage.

[7]Whenever no (reliable) information can be obtained about the degree of support, we safely assume support is missing; $\sigma = 0$.

[8]As an example, tools claiming certain functional capabilities merely by the provision of some general-purpose macro language or *Application Programming Interface* (API) cannot be considered meeting our interest in directly applicable, off-the-shelf tools.

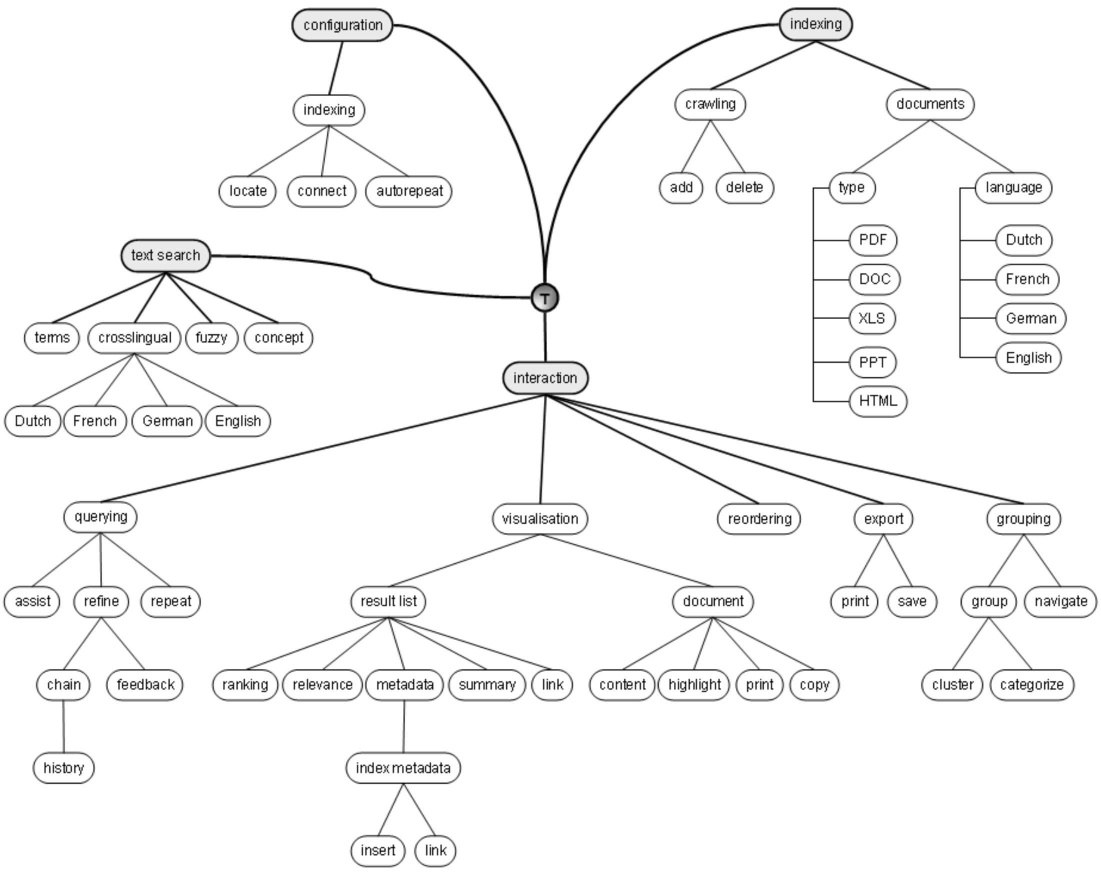

Fig. 2. The requirements tree for use case "free text search". Associated objective functions are not displayed and node labels have been replaced by more descriptive terms.

Third, interpretation of results should primarily be based on a relative comparison of tools by identifying any significant differences in conformity scores, as the absolute scores may depend heavily on the somewhat arbitrary structuring of the tree, composition of objective functions, and parameter settings.

As a last remark, we observed a marked difference in prioritizing functionalities among different user profiles. Whereas some profiles cautiously distributed priorities as if they were given some fixed amount of priority points, others rated the majority of functionalities equally and sufficiently high. Judicious use of normalizing operators in objective functions at different levels in the requirements tree prevents the model from being biased by these different prioritizing behaviours. The successive application of small-scale normalisations will give the desired effect of conformity scores being somewhat more tailored for profiles having defined more balanced priority schemes, provided those scheme reflect actual gradations in desirability of functional needs.

2) Possible uses: Given accurate support values and priorities, one could use this procedure to make a selection of tools on the basis of functional conformity, as suggested by (1). Such selection would allow to identify tools that are promising and suitable candidates for further, more thorough testing. Since the number of tools on the market is usually quite large, and time is limited in research projects, this early kind of preliminary evaluation may turn out to be an interesting, efficient and effective exercise.

As we had little prior knowledge about the tools under study and too little time to perform a full-scale support analysis of the tools, we decided to make a preselection motivated through early conformity impressions drawn from tool documentation, demo presentations and personal contacting, and retaining the conformity evaluation procedure until a later stage of our project, when we will have obtained sufficient hands-on experience, acquaintance with, and knowledge about the tools. This way, conformity scores will be presented alongside the results of other criteria and general findings, such that the tools can be rated according to the entire evaluation spectrum (recall Fig. 1).

IV. Related work

A. Technology

There is a growing interest in crime data mining techniques. Traditional data mining techniques such as association analysis, classification and prediction, cluster analysis, and outlier analysis identify patterns in structured data ([1]). New techniques are capable to identify patterns from both structured and unstructured data sources and become very valuable to apply as crime investigators currently consider various automated data mining techniques for both local law enforcement and national security applications. The following techniques are most commonly studied.

1) Entity extraction: Entity extraction identifies particular patterns from unstructured data sources such as text, images, or audio materials. It has been used to automatically identify persons, addresses, vehicles, and personal characteristics from police narrative reports ([2]). Entity extraction provides basic information for crime analysis, but its performance depends greatly on the availability of extensive amounts of clean input data.

2) Clustering: Clustering techniques group sufficiently similar data items into automatically discovered groups, for example, to identify suspects who conduct crimes in similar ways or distinguish among groups belonging to different gangs. Clustering crime incidents can automate a major part of crime analysis but is limited by the high computational intensity typically required.

3) Classification: Classification finds common properties among different crime entities and organizes them into predefined classes (as opposed to clustering, which is unsupervised). Often used to predict crime trends, classification can reduce the time required to identify crime entities or to sift through large document collections. Classification also requires reasonably complete training and testing data because a high degree of missing data would limit prediction accuracy.

4) Sequential pattern mining: Sequential pattern mining finds frequently occurring sequences of items over a set of transactions that occurred at different times. Showing hidden patterns benefits crime analysis, but to obtain meaningful results it requires rich and highly structured data.

5) Deviation detection: Deviation detection uses specific measures to study data that differs markedly from the rest of the data. Also called *outlier detection*, investigators can apply this technique to fraud detection, network intrusion detection, and other crime analyses.

6) String comparison: String comparator techniques compare the textual fields in pairs of database records and compute the similarity between the records. These techniques can detect deceptive information in criminal records, such as name, address, and Social Security number ([3]). Investigators can use string comparators to analyze textual data, but the techniques often require intensive computation.

7) Social network analysis: Social network analysis ([4]) describes the roles of and interactions among nodes in a conceptual network. Investigators can use this technique to construct a network that illustrates criminals' roles, the flow of tangible and intangible goods and information, and associations among these entities. Further analysis can reveal critical roles and subgroups and vulnerabilities inside the network. This approach enables visualisation of criminal networks.

B. Projects

Important crime data mining projects worldwide include COPLINK ([5]), CLEAR (Citizen Law Enforcement Analysis and Reporting), FLINTS (Forensic Led Intelligence System) ([6]), OVER ([7]), KDD-PN (Knowledge Discovery from Databases Police Netherlands). The Belgian INFO-NS project especially focuses on the evaluation and comparison of text mining tools, when they are used in order to give police forces a better picture and insight in crime and its processes, complementary to any tools which are based on existing structured data.

V. CONCLUSION

The development and deployment of new information technologies in a domain where informatisation has been slowly, seemingly hesitatingly, but nevertheless relentlessly introduced, is a great challenge with many opportunities. Through our own project with the Belgian police, we encountered many interesting aspects that are not readily found or touched upon in literature on the subject, most noticeably on the issues of privacy, security, legal aspects such as the evidential value of generated results, data preprocessing and cleaning, integration, flexibility, adaptability, and performance of exploitation tools in practical settings.

In this paper we presented our proposed evaluation methodology for conformity testing of software tools, which fits in a larger framework of tool evaluation. We hope our work may prove useful, inspire or ponder other field workers on these topics, as we believe the success and promosing future of these tools heavily depends on their careful consideration.

ACKNOWLEDGMENT

The authors would like to thank the Belgian police for their interest and active collaboration, in particular Kris D'Hoore, Martine Pattyn and Paul Wouters. This work was supported by the Belgian Science Policy Office through their AGORA research programme. AG/01/101

REFERENCES

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufman, 2001.

[2] M. Chau, J. J. Xu, and H. Chen, "Extracting meaningful entities from police narrative reports," in *Proceedings of the National Conference for Digital Government Research*, vol. 2, May 2002.

[3] G. Wang, H. Chen, and H. Atabakhsh, "Automatically detecting deceptive criminal identities," *Communications of the ACM*, vol. 47, no. 3, 2004.

[4] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[5] R. V. Hauck, J. Schroeder, and H. Chen, "Coplink: Developing information sharing and criminal intelligence analysis technologies for law enforcement," in *Proceedings of the National Conference for Digital Government Research*, vol. 1, May 2001, pp. 134–140.

[6] R. M. Leary, "The role of the national intelligence model and flints in improving police performance," Online at http://www.homeoffice.gov.uk/docs2/resconf2002daytwo.html.

[7] G. C. Oatley, B. W. Ewart, and J. Zeleznikow, "Decision support systems for police: Lessons from the application of data mining techniques to 'soft' forensic evidence," *Artificial Intelligence and Law*, (Submitted).