Krassimir Markov, Vladimir Ryazanov,
Vitalii Velychko, Levon Aslanyan
(editors)

New Trends in Classification and Data Mining

I T H E A
SOFIA
2010

Krassimir Markov, Vladimir Ryazanov, Vitalii Velychko, Levon Aslanyan (ed.)

New Trends in Classification and Data Mining

ITHEA®

Sofia, Bulgaria, 2010

First edition

Recommended for publication by The Scientific Concil of the Institute of Information Theories and Applications FOI ITHEA

This book maintains articles on actual problems of classification, data mining and forecasting as well as natural language processing:

- new approaches, models, algorithms and methods for classification, forecasting and clusterisation. Classification of non complete and noise data;
- discrete optimization in logic recognition algorithms construction, complexity, asymptotically optimal algorithms, mixed-integer problem of minimization of empirical risk, multi-objective linear integer programming problems;
- questions of complexity of some discrete optimization tasks and corresponding tasks of data analysis and pattern recognition;
- the algebraic approach for pattern recognition - problems of correct classification algorithms construction, logical correctors and resolvability of challenges of classification, construction of optimum algebraic correctors over sets of algorithms of computation of estimations, conditions of correct algorithms existence;
- regressions, restoring of dependences according to training sampling, parametrical approach for piecewise linear dependences restoration, and nonparametric regressions based on collective solution on set of tasks of recognition;
- multi-agent systems in knowledge discovery, collective evolutionary systems, advantages and disadvantages of synthetic data mining methods, intelligent search agent model realizing information extraction on ontological model of data mining methods;
- methods of search of logic regularities sets of classes and extraction of optimal subsets, construction of convex combination of associated predictors that minimizes mean error;
- algorithmic constructions in a model of recognizing the nearest neighbors in binary data sets, discrete isoperimetry problem solutions, logic-combinatorial scheme in high-throughput gene expression data;
- researches in area of neural network classifiers, and applications in finance field;
- text mining, automatic classification of scientific papers, information extraction from natural language texts, semantic text analysis, natural language processing.

It is represented that book articles will be interesting as experts in the field of classifying, data mining and forecasting, and to practical users from medicine, sociology, economy, chemistry, biology, and other areas.

General Sponsor: Consortium FOI Bulgaria (www.foibg.com).

Printed in Bulgaria

ISBN 978-954-16-0042-9

C\o Jusautor, Sofia, 2010

NATURAL INTERFACE TO ELECTION DATA

Elena Long, Vladimir Lovitskii, Michael Thrasher

Abstract: *Modern technology has the facility to empower citizens by providing easy access to vital electoral information. The majority of such users simply want to use the information; they do not wish to become embroiled in technological details that provide that access; the technology is a means to an end and if allows to obscure the real purpose (to access information) it represents a cost not a benefit. Much of the potential benefit is therefore lost unless a simple and consistent interface can be provided which shields the user from the complexity of the underlying data system retrieval and should be natural enough to be used without training. Currently there are limited tools and information available online where end users can view and interrogate electoral data. The main purpose of our paper is to report upon developments that seek to provide an easy to use interface for users to obtain information regarding the results of general elections within the United Kingdom (UK).*

Keywords: *natural interface, natural language processing, database accessing, SQL-query, production rules*

ACM Classification Keywords: *I.2 Artificial intelligence: I.2.7 Natural Language Processing: Text analysis.*

Introduction

Following the rapid development of both computer and communications technologies, our society now has the potential to access vast amounts of information almost instantaneously on a world-wide basis. One of the major obstacles to achieve it is to realising the potential for wealth and knowledge creation that information represents is the means of simple access by naïve users to relevant information locked in possibly complex data structures. Individuals are not expected to know in detail what information is required, or where it might be found and certainly does not know about data structures. In respect of electoral data, for example, the citizen simply requires information that is relevant to his or her particular area of interest - and no more.

This paper represents results of our further research in the natural language interface creation to database (DB) [V.A.Lovitskii and K.Wittamore, 1997; Guy Francis *et al.*, 2007; Elena Long *et al.*, 2009]. The data source addressed here is the DB with the results of 2005 UK General Election. The result is our current vision of *"natural interface"* has been implemented (http://141.163.170.152:8080/NITED/NITEDJSP.jsp) as a Web application named **NITED** (**N**atural **I**nterface **T**o **E**lection **D**ata) where a user can see essential election data online. The aim of design is that the application must offer simple, intuitive and responsive user interfaces that allows users to achieve their objectives regarding information retrieval with minimum effort and time..

Despite the intuitive appeal of a natural language interface, some researchers have argued that a language like English has too many ambiguities to be useful for communicating with computers. Indeed, there is little experimental data supporting the efficacy of a natural language interface, and the few studies that have compared natural language interfaces to other styles of interface have been generally negative towards the former.

Indeed, two major obstacles lie in the way of achieving the ultimate goal of support for arbitrary natural language queries. First, automatically understanding natural language (both syntactically and semantically) remains an open research problem. Second, even if there were a perfect parser that could fully understand any arbitrary natural language query, translating the parsed natural language query into a correct formal query still remains an issue since this translation requires mapping the understanding of intent into a specific database schema.

Natural language is not only very often ambiguous but is dependent on a great deal of world knowledge. In order to implement a working natural language system one must usually restrict it to cover only a limited subset of the vocabulary and syntax of a full natural language. This allows ambiguity to be reduced and processing time to be kept within reasonable bounds. In order for it still to be considered a natural language interface, most of the positive traits of a general natural language interface would have to be maintained. To retain the properties of ease of use and ease of remembering, the limitations of the system must somehow be conveyed to the user without requiring them to learn the rules explicitly.

Natural language interfaces, if they are the only form of interaction, do not take advantage of the capabilities of the computer -- those strategies that work in human-human communication are probably not best suited to human-computer interactions, where the computer can display information many times faster than people can enter commands

The principal purpose of our paper is to offer the natural (versus natural language) user interface which makes it easy, efficient, and enjoyable to operate NITED in a way which produces the desired result. This generally means that the user is required to provide minimal input to achieve the desired output, and also that NITED minimizes undesired outputs or data clutter.

Reading this paper will tell you the following:

• Natural user interface.

• Natural user enquiry.

• Help instructions.

• Production rules.

• Natural enquiry to SQL query conversion.

Natural User Interface

The natural user interface (NUI) is a key to application usability. NUI is needed when interaction between users and NITED occurs. The goal of interaction between the user and the NITED at the NUI is effective operation and control of the NITED, and feedback from the NITED in desirable for the user format i.e. NUI provides a means of input, allowing the users to ask question, and output, allowing the NITED to reply on user's question.

The design of a NUI affects the amount of effort the user must expend to provide input and to interpret the output of the system, and how much effort is required to learn this. Usability is mainly a characteristic of the NUI, but is also associated with the functionalities of the product and the process to design it. It describes how well the NITED can be used for its intended purpose by its target users with efficiency, effectiveness, and satisfaction, also taking into account the requirements from its context of use. A key property of a good user interface is consistency.

There are three important aspects [http://en.wikipedia.org/wiki/User_interfac] to be taken into account. First, the controls for different features should be presented in a consistent manner so that users can find the controls easily. For example, users find it very difficult to use software when some commands are available through menus, some through icons, and some through right-clicks. A good user interface might provide shortcuts or "synonyms" that provide parallel access to a feature, but users do not have to search multiple sources to find what they're looking for.

Second, the "principle of minimum astonishment" is crucial. Various features should work in similar ways. For example, some features in Adobe Acrobat are "select tool, then select text to which apply." Others are "select text, then apply action to selection.

Third, user interfaces should strive for minimum change version-to-version -- user interfaces must remain upward compatible. For example, the change from the menu bars of Microsoft Office 2003 to the "ribbon" of Microsoft Office 2007 is universally hated by established users, many of whom found it difficult to achieve what had become routinized tasks. The "ribbon" could easily have been "better" in the mid-1990's than the menu interface if writing on a blank slate, but once hundreds of millions of users are familiar with the old interface, the costs of change and adaptation far exceed the benefit of improvement. The vast majority of users viewed this forced change, without a backward-compatibility mode, as unfavorable; more than a few viewed it as verging on malevolence. Re-design should introduce change incrementally such that existing users are not alienated by a revised product.

Good user interface design is about setting and meeting user expectations because the best NUI from a programmer's point of view is not, as a rule, the best from a user's point of view.

We have tried to create a NUI to improve the efficiency, effectiveness, and naturalness of user-NITED interaction by representing, reasoning, and acting on models of the user, domain and tasks. The main part of NUI is a graphical interface, which accepts input via computer keyboard and mouse. The actions are usually performed through direct manipulation of the graphical control elements. The natural way to represent the output for election application domain (EAD) is a table. In the next section we will discuss in detail the input enquiry presentation.

Natural User Enquiry

- Over a number of years [Guy Francis *et al.*, 2007; Elena Long *et al.*, 2009] users' natural language enquiries (NLE) have been collected by us in a series of research programmes. Direct observation of users' NLE shows, unsurprisingly, that all users are lazy i.e. they want to achieve the desired result whilst expending minimum effort. They do not want to type in the long NLE such as "*How many votes did the Demanding Honesty in Politics and Whitehall candidate obtain in Dumfriesshire, Clydesdale and Tweeddale"?* This is the natural behaviour of human being in accordance with the ***principle of simplicity***, or ***Occam's razor principle*** *(Occam's (or Ockham's) razor is a principle attributed to the 14th century logician and Franciscan friar; William of Occam. Ockham was the village in the English county of Surrey where he was born)*. The principle states that "Everything should be made as simple as possible, but not simpler". Finding a balance between simplicity and sophistication at the input side has been discussed elsewhere [L.Huang et al., 2001].

On the one hand, NLE provides end users with the ability to retrieve data from a DB by asking questions using plain English. But, on the other hand, there are several problems of using NLE:

- The end users are generally unable to describe completely and unambiguously what it is they are looking for at the start of a search. They need to refine their enquiry by giving feedback on the results of initial search e.g. *"I'm looking for a* ***nice*** *city in France for holiday"* (where *Nice* is a city in France but also an adjective in English). Similar ambiguities exist for the UK general election database. For example, the words *Angus, Bath, Corby, …, Wells* are values of fields *Constituency* and *Surname in the General Election data 2005 DB* but are alos common nouns and place names. Parsing of such simple NLE is quite complicated and requires powerful knowledge base from system [V.A.Lovitskii and K.Wittamore, 1997].

- It is simply impossible to require that users know the exact values in DB (e.g. name of constituency). For example, if user makes the enquiry: "*Who won the election in Suffolk Central & Ipswich North"?* but instead of using the symbol ***'&'*** types in ***"and"*** NITED will not find the constituency in DB.
- In the case when user simply made a mistake and instead of typing in the desirable constituency *Hereford* in the NLE: "*Who won the election in Hereford"* user entered *Hertford* (it's **wrong** but at the same time it's **right** from the NITED point of view because it has the right part of an existing constituency *Hertford & Stortford*), NITED will find the answer for the constituency *Hertford & Stortford*. When user sees the response, he/she realises that constituency was wrong and simply corrects it.
- As a rule a user's NLE cannot be interpreted by NITED without additional knowledge because the concepts involved in NLE are outside of the EAD. For example, in NLE *"How did the Conservative **perform** in South West?"* NITED should know the meaning of word *"perform"* regarding the election data, and in the NLE *"Which party **won** the Aberdeenshire West and Kincardine constituency?"* correctly interprets word "won".
- In conclusion it would be sensible to underline the main problem which hinders the use of NLE the cognitive process of "understanding" is itself not understood. First, we must ask: "*What it means to understand a* NLE?" The usual answer to that question is to model its meaning. But this answer just generates another question: "*What does meaning mean?"* The meaning of a NLE depends not only on the things it describes, explicitly and implicitly, but also on both aspects of its causality: "*What caused it to be said*" and "*What result is intended by saying it*". In other words, the meaning of a NLE depends not only on the sentence itself, but also on the context: **Who** is asking the question, and **How** the question is phrased.

In the result of NLE analysis we decided to distinguish two different types of NUE: (1) NLE **T**emplate (**NLET**) and (2) **N**atural **D**escriptors **E**nquiry (**NDE**). Such enquiries permit users to communicate with a DB in a natural way rather than through the medium of formal query languages. Obviously issues in these two NUE are related, and the knowledge needed to deal with them is represented as a set of Production Rules (PR). Let us consider these two types of NUE.

Natural Language Enquiry Template combines a list of values to be selected when required and generalization of users' NLETs. Examples of some **F**requently **A**sked **Q**uestions (**FAQ**) are shown below:

- What was the **result** in [constituency]?
- In which **constituency** did [party] achieve its **highest vote**?
- **Who won** the [constituency]?

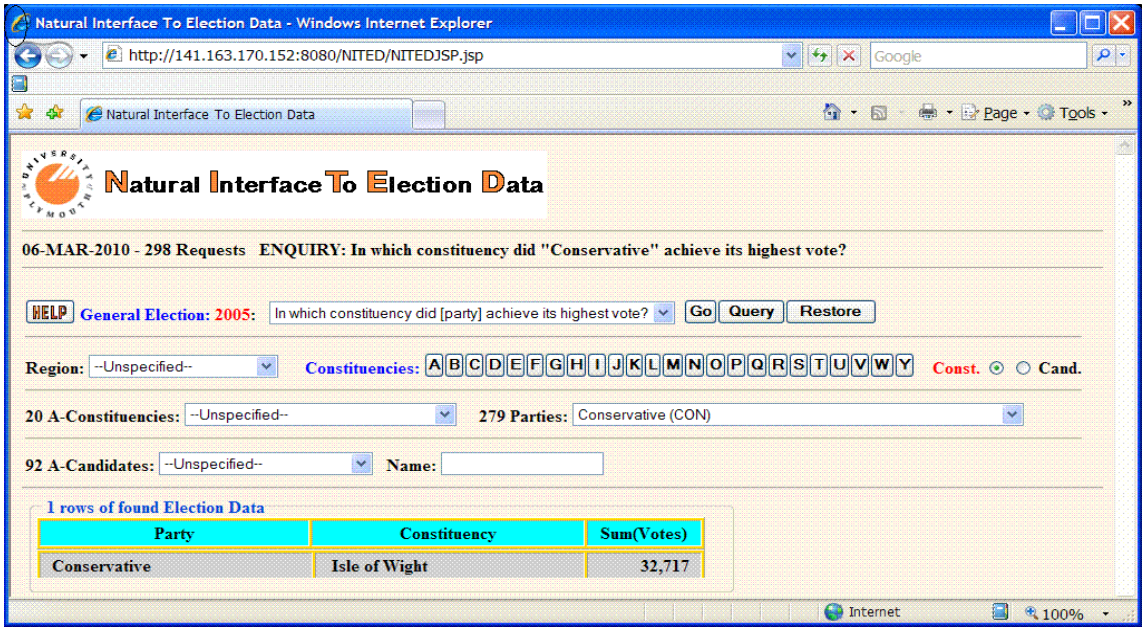

Figure 1. Natural Language Enquiry Template

The initial set of FAQ has been created by export in *EAD* but in the result of activities new NUE have been collected by NITED, analysed, generalized, converted to the NLET. These have then either been added to FAQ, or substituted for the under-used NLET. When the user selects an appropriate NLET with some descriptor in square brackets, selects the corresponding values from the list and click button ***Go*** the result will be displayed instantly (see Figure 1).

The user can build his/her own enquiries using any combination of the descriptors, each of them represents the corresponding meaningful field of the Election DB (see Figure 2). The definition of "meaningful fields" depends on AD objectives. For the considered EAD is a list of descriptors: {*region, constituency, party,* etc.}. Between descriptors and meaningful fields exist one-to-one attitude. Such attitudes are represented by the production rules (see section below).

Let's call enquiries using descriptors as a **Natural Descriptors Enquiries** (NDE). For example, if user wants a list of all the women elected in the South West region simply click the following check boxes: "Party", "Candidate", "Votes", "Sits in Parliament" and radio button "Female". Then select the South West region from the drop down menu of regions. When NDE is ready the user should simply click "Go" button and NITED instantly displays the result (see Figure 2). As user clicks the check boxes and selects the radio buttons NDE appears in the space next to the date above. If user clicks a check box but then change his/her mind the check box should simply be clicked again.

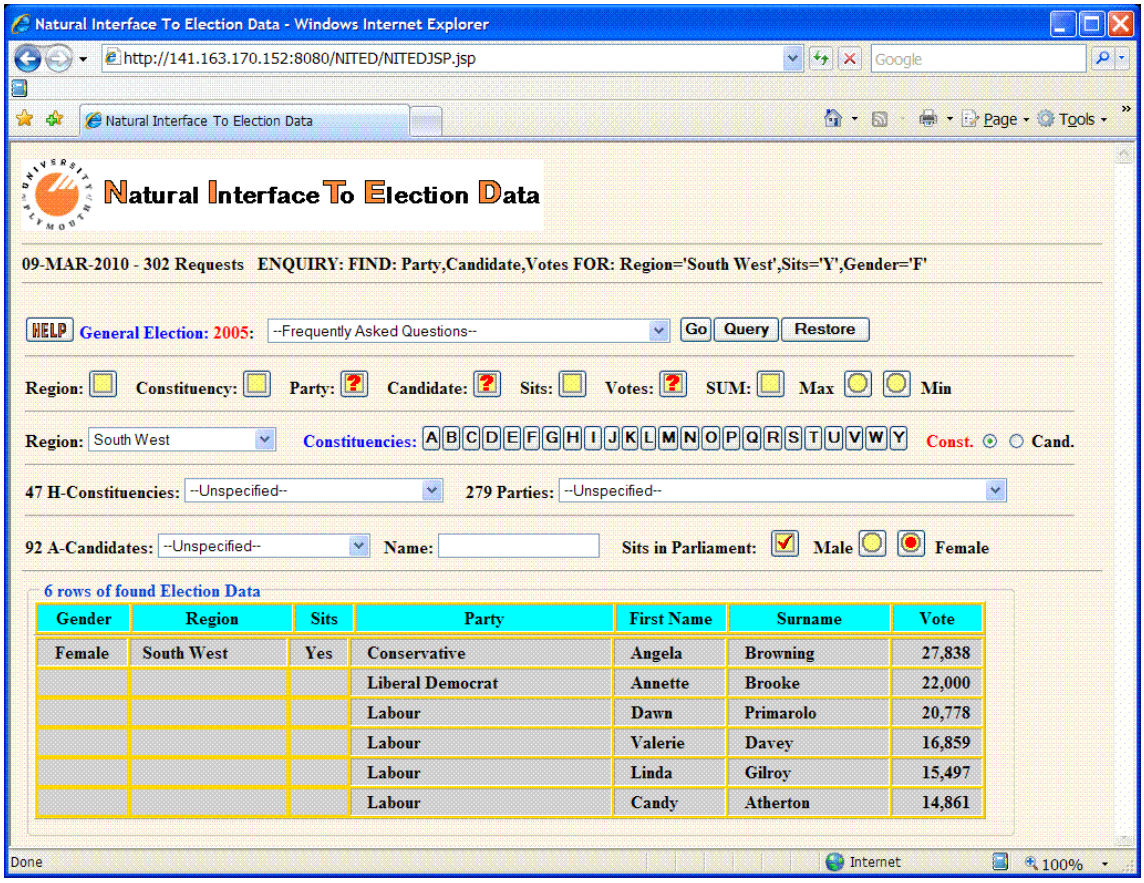

Figure 2. Natural Descriptors Enquiry

Help Instructions

Help Instructions (HI) in a Web application context means on-screen help. HI are needed for system efficiency and users' satisfaction. Clear HI can significantly reduce the number of disappointed users. Producing clear instructions that really help people is difficult as evidenced by the low-quality instructions encountered in many web applications. If designing HI were easy, there would not be so many poor examples!

Good HI have to take into account the type of users who will presumably use the NITED:

- Users' computer literacy is the basic IT literacy.
- Users should not require a conceptual background before they can use the NITED.
- Users might be absolute beginners or moderately familiar with the subject but they should not be subject matter expert.

Requirements to Help Instructions:

- HI should be short enough but provide sufficient information about the screen function.
- Good HI does not mean that all options should be explained in detail.
- HI should include brief information that is at least sufficient to get started.
- The most frequently used features should be explained.
- Top-level tasks, without much detail about particular fields, should be described.
- Step-by-step worked examples that users can follow should be represented in the .HI.

We tried to meet all of these requirements in the HI for NITED (see Figure 3).

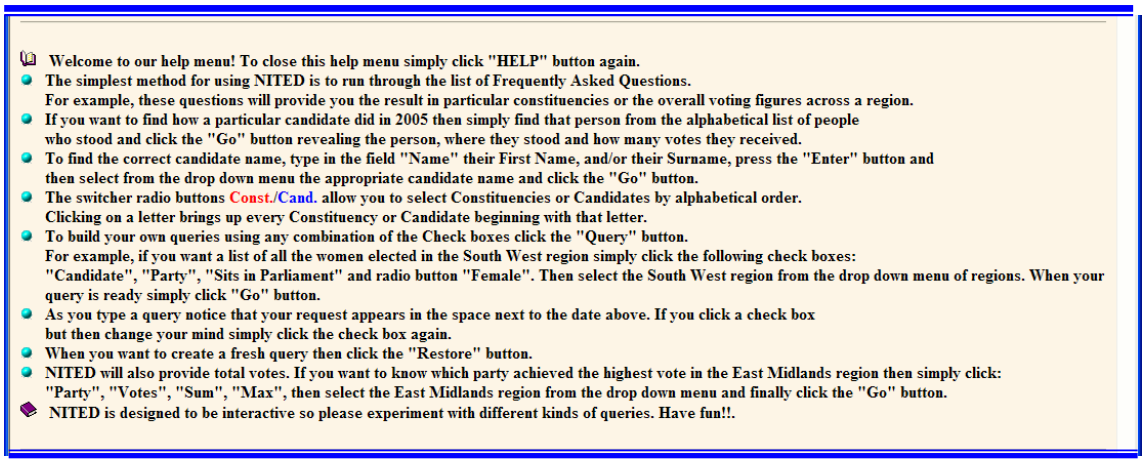

Figure 3. Help Instructions

Production Rules

At first glance, the NLET is an ideal way to communicate with EAD but in reality there are some problems, which need to be solved to provide lightness of communication. To highlight such problems is enough to consider quite a simple NLET: "*Who won an election in [constituency]?"* or *"How did the [party] perform in [region]?"*. Without knowing *"who is who"* and meaning of *"won"* and *"perform"* NITED cannot answer such questions. To explain it to NITED the **P**roduction **R**ules (**PR**) need to be involved. Many researchers are investigating what information is needed and how the information needs to be represented in the PR. From our point of view the **P**reconditioned PR (PPR) should be used. The PPR is a quite powerful approach to solve this problem. The subset of PPR in format:

<Precondition> ↦ <Antecedent> ⇒ <Consequent>

is shown below.

1. AD:Election2005 ↦ who ⇒ candidate;
2. AD:Election2005 ↦ [candidate]:<win⊕won⊕highest > ⇒ [SQL]:<MAX(votes)>;
3. AD:Athletics ↦ [runner]:<win⊕won> ⇒ [SQL]:<MIN(time);
4. AD:Athletics ↦ [shooter]:<win⊕won> ⇒ [SQL]:<MAX(distance);
5. AD:Election2005 & DB:MS Access ↦ votes ⇒ [Field]:<gcr_post_election_votes>;
6. AD:Election2005 & DB:MS Access ↦ candidate ⇒ [Field]:<can_first_name, can_last_name>;
7. AD:Election2005 & DB:Oracle ↦ [party]:<win⊕won⊕highest> ⇒ [SQL]:<MAX(SUM(votes))>;
8. AD:Election2005 & DB:MS Access ↦ [party]:<win⊕won⊕highest > ⇒ [SQL]:<TOP1, SUM(votes),DESC>;
9. AD:Election2005 & DB:MS Access ↦ perform ⇒ candidate,votes;

where ⊕ **-** denotes "*exclusive OR*"**. Precondition** consist of **$class_1$:$value_1$ {& $class_i$:$value_i$}**. **Antecedent** might be represented by: (i) **single word** (e.g. *who, won, perform, etc.*), (ii) **sequence of words** (e.g. *as soon as*,

create KB, *How are you doing*, etc.), or (iii) **pair - [context]:<value>**. Context allows one to avoid word ambiguity and thereby distinguish difference between "*Candidate won an election*" and "*Party won an election*". Presentation of **Consequent** is similar to Antecedent structure except (iii). For Consequent pair represents **[descriptor]:<value>**.

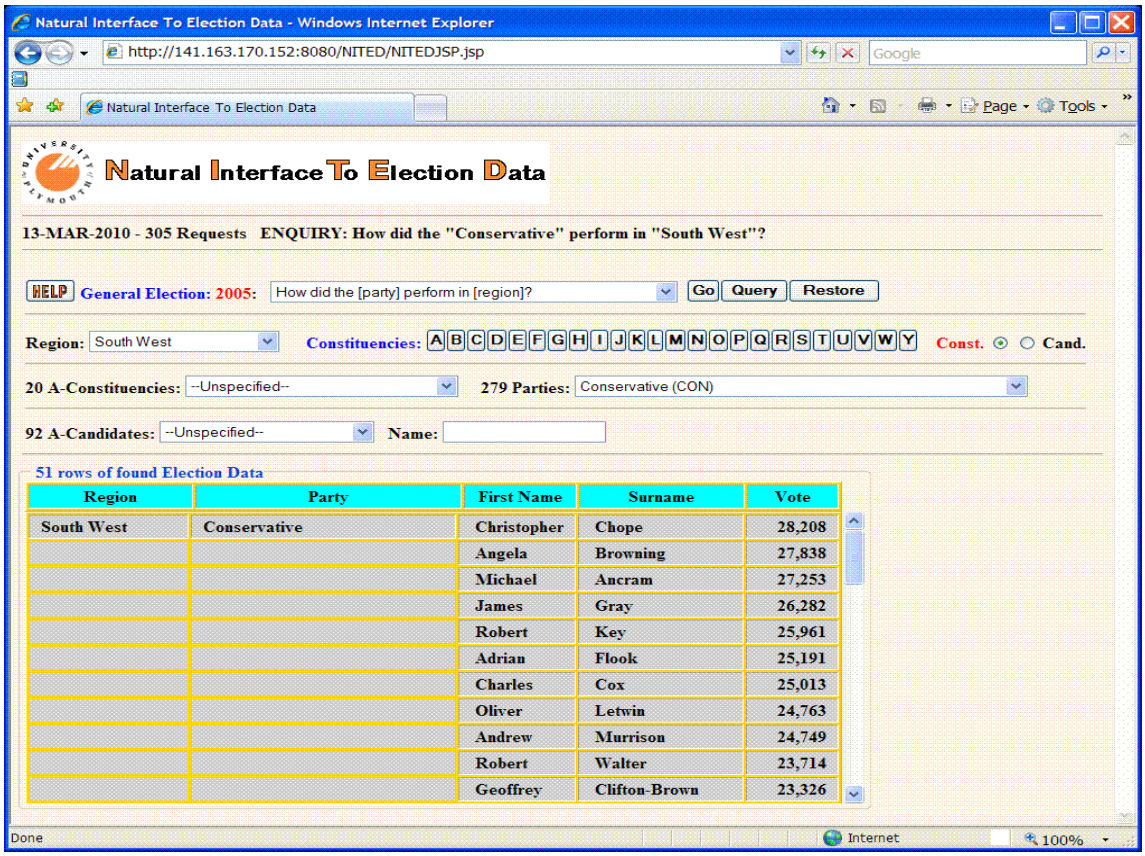

Figure 4. Reply to NLET after describing the word *"perform"* in the PPR

For EAD subset {1, 2, 5, 6, 8, 9} of PPR is used. PPR 3 and 4, in fact, show another meaning of the same word *"won" but* for a different AD. The PPR 7 shows the simplest way to cover the difference in SQL for different DB. Result of using selected PPR to reply to NLET *"How did the [party] perform in [region]?"* is shown on Figure 4.

Thus, NLET allows the user to "be lazy" but requires some effort to create the proper set of PPR.

Natural Enquiry to SQL Query Conversion

Two types of NUE have been considered. The NDE does not require great effort to be converted to the corresponding SQL query. Only NLET need some parsing. The mechanism of NLET parsing is very simple: *"eliminating the unnecessary until only the necessary remains"*. Several steps involved in NLET processing.

- NITED takes the NLET as a character sequence and converts the original NLET to a *skeleton* by noisy (non-searchable) words elimination. As a result of such conversion the NLET will contain only **meaningful** words: let's call the word meaningful if it represents DB field descriptor or DB field value.
- EAD is represented by DB. DB **meaningful** fields (i.e. they don't represent primary or foreign keys) contain election data. Each meaningful fields has a list of descriptors. Between descriptors and meaningful fields exists an one-to-one attitude.

- The purpose of NLET processing is to match NLET meaningful words against the DB fields descriptors.
- The final step of NLET to SQL query conversion is rather complicated because it is necessary to access data from many different tables within an EAD and join those tables together in SQL query.

Conclusion

NITED is designed through the Internet to make nationwide election results available to any user. We hope that NITED has the potential to change certain aspects of political behaviour, including people's desire to engage with the political process. Like any technology, systems like NITED can have a wide variety of effects on political behaviour and practices, but it is too soon yet to make general conclusions about its impact. Nevertheless, we intend that, following the 2010 UK General Elections in the NITED will play an important role, helping to make nationwide election results available to Web users.

Bibliography

Guy Francis, Mark Lishman, Vladimir Lovitskii, Michael Thrasher, David Traynor, 2007. "Instantaneous Database Access", *International Journal "Information Theories & Applications", Vol 14(2)*, 161-168.

L.Huang, T.Ulrich, M.Hemmje, E.Neuhold, 2001. "Adaptively Constructing the Query Interface for Meta Search Engines", Proc. of the Intelligent User Interface Conf.

Elena Long, Vladimir Lovitskii, Michael Thrasher, David Traynor, 2009. "Mobile Election", International Book Series "Information Science and Computing", Book 9, Intelligent Processing, 19-28.

V.A.Lovitskii and K.Wittamore, 1997. "DANIL: Databases Access using a Natural Interface Language", P*roc. of the International Joint Conference on Knowledge-Dialogue-Solution: KDS-97*, Yalta (Ukraine), 282-288.

Authors information

Elena Long – University of Plymouth, Plymouth, Devon, PL4 6DX, UK,
e-mail: elena.long@plymouth.ac.uk
Major Fields of Scientific Research: Political science

***Vladimir Lovitskii** – University of Plymouth, Plymouth, Devon, PL4 6DX, UK,*
e-mail: vladimir.lovitskii@fsmail.net.
Major Fields of Scientific Research: Artificial Intelligence

***Michael Thrasher** – University of Plymouth, Plymouth, Devon, PL4 6DX, UK*
e-mail: mthrasher@plymouth.ac.uk
Major Fields of Scientific Research: Political science