Automatic mark-up of legislative documents and its application to parallel text generation

Lorenzo Bacci[1], Pierluigi Spinosa[1], Carlo Marchetti[2,3], and Roberto Battistoni[3]

[1] Institute of Legal Information Theory and Techniques,
Florence, Italy
bacci@ittig.cnr.it
spinosa@ittig.cnr.it
[2] Dipartimento di Informatica e Sistemistica dell'Università di Roma "La Sapienza"
carlo.marchetti@senato.it
[3] Italian Senate, Rome, Italy
r.battistoni@senato.it

Abstract. In the juridical domain a huge amount of plain legislative acts have been produced since and before the advent of computers and word processors. The conversion of legacy and plain documents in a standard XML format implies great and numerous benefits. In order to accomplish this task, several automatic and semi-automatic tools have been developed in the last ten years. In this paper, xmLegesMarker, developed at Ittig[4], the Italian state-of-art legislative documents parser, is presented. The tool is NIR standard compliant, it's embedded in xmLegesEditor and it was recently adopted for evaluation by the Italian Senate in order to automatically spawn the parallel text (*testo a fronte*), i.e. the document used to highlight the modifications introduced during the debate of a bill.

1 Introduction

xmLegesMarker is a structure parser for legislative documents. Its development started in 2003, within Norme In Rete (NIR) project[1], in order to provide a detailed mark-up of legacy documents and, generally, plain legislative acts. Although it was realized as a stand-alone software, it has been mainly exploited inside xmLegesEditor[2], an XML based legislative editor arisen from the NIR project, to implement the import function, namely the migration of plain texts into the XML environment. The NIR project defined also appropriate DTD and XMLSchema, which represent the basis of both xmLegesEditor and xmLegesMarker and aim at describing in a very detailed way the Italian legislative acts. Thanks to its independent development and the ongoing improvements, xmLegesMarker has also been effectively adopted in the last few years by several public administrations and regional governments, usually to pursue a migration of their plain text databases of local, regional and national laws towards the NIR-XML mark-up.

[4] Institute of Legal Information Theory and Techniques (CNR)

Several benefits derive from this process: structured documents enhance information retrieval and normative system maintenance, and can represent the ground for a further semantic description of text in terms of provisions [3], but, more important, they can also be exploited for legislative-domain strictly-related operations. As an example, TafWeb, described in section 5, is a smart legislative text comparison software, developed by the Computer Science Department of the University of Bologna under the supervision of the Italian Senate, which exploits the XML mark-up of the versions of a bill under debate in order to generate the parallel text document (*testo a fronte*). The Italian Senate is currently evaluating xmLeges-Marker within the TafWeb suite in order to automatize the production of the first version of a parallel text, starting from the plain Chamber and Senate versions of the bill.

2 Approaching plain documents

Automatically structuring a plain document means creating a software that, given the plain document as input, is able to assign an identity to every piece of text. In the XML world, this task is accomplished by putting the text between tags. In this case, the marker uses NIR defined tags in order to create well formed and valid, with respect to NIR schema, outputs.

Generally speaking, the information that can be obtained from a plain document consists of data and meta-data. In legislative acts, the enacting terms section, in which articles and paragraphs lie, matches the data, while entities like title, number and type of document, subscribers and so on, are considered explicit meta-data. Other meta-data, defined in NIR schema, although not explicitly present in the input, are computed or added by xmLegesMarker, usually exploiting the values of the explicit meta-data (i.e.: automatic generation of URN [4]). The splitting of information in data and meta-data follows the physical structure of the document. While explicit meta-data are typically located in the header or in the footer, the body of the legislative act accommodates the enacting terms, namely the data.

Besides the physical position, there is another important difference between the body and the header (or the footer) in a legislative act: the former is composed by partitions strictly organized and sorted in a hierarchical way, practically a tree of partitions, while the latter appears fuzzy and composed by expected and unexpected elements, often in a random order. This is the reason why xmLegesMarker adopts two different strategies in order to analyze the header, the footer and the body of a document.

2.1 Header and footer

The fuzziness that belongs to both the header and the footer of a legislative act requires a statistical and machine learning approach for meta-data extraction. In order to accomplish this, the theory of Hidden Markov Model (HMM) [5] was chosen and a model, able to understand the most typical information lying in the header and footer of a generic legislative act, was developed. [6]

An important source of information which was exploited to improve the accuracy of the header analysis is represented by the legislative document subtype (act, bill, decree, regional act, etc.). Depending on the subtype in fact, more precise patterns stand out. Therefore, xmLegesMarker applies a specific HMM model if the input subtype is known and supported, the generic model otherwise. Using subtype-crafted models brings two benefits:

- a better formalization of the most important subtype, reaching higher (far higher in some cases, like bills) degrees of accuracy;
- a guess about the subtype when the subtype information isn't given, just applying all the specific model and checking which one fits better.

2.2 Body

The body of a legislative act coincides with the enacting terms section, which is typically well organized in known partitions, hierarchically arranged in a tree structure. On the other hand, the tree representing the enacting terms can be complex, long and nested. An automata approach is required in order to efficiently parse this kind of structure. The Flex scanner generator[5], which has been used to accomplish several tasks for and besides body analysis, allows the creation of very powerful text scanner based on a non deterministic finite state automata (NFA). [6]

The automata that handles the enacting terms strictly follows the constraints imposed by the NIR schema: the parsing process obeys to rules that depend on the automata states (start conditions), which match all the partitions defined in the NIR hierarchy. For example, an alphabetical list can be read only if the automata is in the paragraph state, because, according to NIR and to legislative drafting rules, a list should only stay inside paragraphs.

2.3 Annexes

Let's conclude this survey on plain legislative documents by describing the way the marker handles annexes. In the legislative domain, annexes are very common: they belong to the legislative document itself, just following the, let's say, main act; they can be simple tables, reporting details, prices, fees, or be legislative documents themselves. Their importance sometimes exceeds even the importance of the main act: for example, there are bills containing just one or two paragraph, followed by a legislative decree containing dozens of articles.

The xmLegesMarker strategy comprises the preliminary segmentation of the input into main act and annexes, and the iteration of the header, body and footer parsing functions, described in the previous sections, on the main act as well on every single annex. In this way, possible legislative documents following the main act receive the same treatment and come out completely marked-up.

[5] http://dinosaur.compilertools.net/flex/

3 A glance in depth

In order to better understand the working, the capabilities and the potentiality of the marker, in this section the most interesting features and issues are deepened and discussed. After a brief overview about the shape of the input, we focus on how the marker handles partitions and amendments, which factors determine an increase of complexity during the body analysis process and how syntactical errors in the source have been successfully tackled.

3.1 Input

The input of xmLegesMarker is a plain legislative act in txt, doc, html or pdf format. The marker is able to manage different subtype of plain Italian legislative document, like act, bill, decree, regional act, etc. As discussed, different weights and models are used for the header and footer analysis depending on the subtype.

3.2 Handling partitions

Partitions are used to structure the fragments of the legislative document body. They are arranged in a hierarchical way. The paragraph is the partition that effectively contains the text of the law, while the greater order partitions (article, chapter, section, etc.) can be seen as containers. Even paragraphs can have sub-partitions: lists, which can be alphabetical, numerical or bulleted. Thanks to the use of regular expression, layout reasoning and the application of NIR constraints, as we have seen in 2.2, the marker is able to identify all of these partitions.

Furthermore, xmLegesMarker performs a check concerning partitions numbering, which turns out to be quite useful:

- as a mean to better identify the next partition, often avoiding ambiguity;
- to assign a unique value to the “id” attribute, defined by NIR, in order to permit referencing to every single partition.

3.3 Dealing with textual amendments

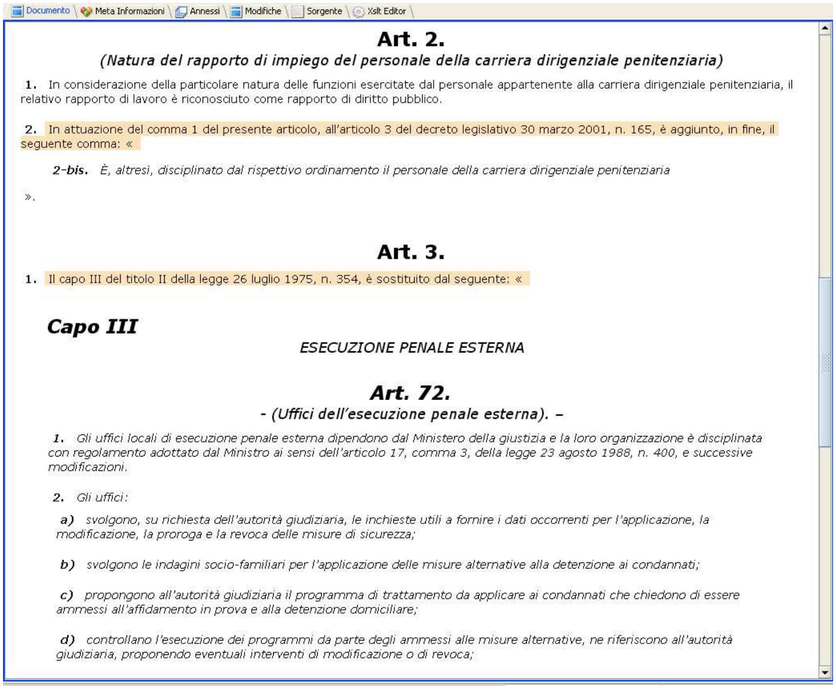
Amendments are a widely used mechanism to express modifications from a legislative document to another legislative document. The textual amendments can be briefly categorized in repeal, insertion and modification, and they can act on words as well as on whole partitions (structural amendment). Insertion and modification amendments are typically expressed using quotes. So, for example, it’s possible to express through an amendment the substitution of a single noun or the insertion of a brand-new article in a precise position of a regional act.

The marker is able to identify and handle both words and structural amendments, and, in case of structural amendment, it enters the amendment and precisely mark-up all the partitions and data contained.

3.4 Increasing complexity

Even though the body of legislative documents generally follows the same syntactical rules, there are several variables that increase the complexity of the automata:

- paragraphs are sometimes not numbered, this happens especially with legacy documents;
- almost every partition can have a partition title, the *rubrica*, that sometimes is placed just after the declaration of the new partition, sometimes below, sometimes it's included between particular separator characters, sometimes not;



Art. 2.

(Natura del rapporto di impiego del personale della carriera dirigenziale penitenziaria)

1. In considerazione della particolare natura delle funzioni esercitate dal personale appartenente alla carriera dirigenziale penitenziaria, il relativo rapporto di lavoro è riconosciuto come rapporto di diritto pubblico.

2. In attuazione del comma 1 del presente articolo, all'articolo 3 del decreto legislativo 30 marzo 2001, n. 165, è aggiunto, in fine, il seguente comma: «

2-bis. *È, altresì, disciplinato dal rispettivo ordinamento il personale della carriera dirigenziale penitenziaria*

».

Art. 3.

1. Il capo III del titolo II della legge 26 luglio 1975, n. 354, è sostituito dal seguente: «

Capo III

ESECUZIONE PENALE ESTERNA

Art. 72.

- (Uffici dell'esecuzione penale esterna). –

1. *Gli uffici locali di esecuzione penale esterna dipendono dal Ministero della giustizia e la loro organizzazione è disciplinata con regolamento adottato dal Ministro ai sensi dell'articolo 17, comma 3, della legge 23 agosto 1988, n. 400, e successive modificazioni.*

2. *Gli uffici:*

a) *svolgono, su richiesta dell'autorità giudiziaria, le inchieste utili a fornire i dati occorrenti per l'applicazione, la modificazione, la proroga e la revoca delle misure di sicurezza;*

b) *svolgono le indagini socio-familiari per l'applicazione delle misure alternative alla detenzione ai condannati;*

c) *propongono all'autorità giudiziaria il programma di trattamento da applicare ai condannati che chiedono di essere ammessi all'affidamento in prova e alla detenzione domiciliare;*

d) *controllano l'esecuzione dei programmi da parte degli ammessi alle misure alternative, ne riferiscono all'autorità giudiziaria, proponendo eventuali interventi di modificazione o di revoca;*

Fig. 1. A pretty nested example visualized in xmLegesEditor: the first paragraph of article 3 contains a chapter substitution amendment.

- within a paragraph there are three allowed kind of lists: alphabetical, numerical and bulleted list; however, good drafting rules state that bulleted lists should be avoided, because they can't be directly referenced (the relative position has to be specified), and only alphabetical lists should stay immediately inside paragraphs while numerical lists should only lie within alphabetical lists;

- parsing the text of amendments sometimes turns out to be a pretty hard task: between quotes there's a no man's land where most of the rules that usually guide the automata don't apply anymore, while every kind of partition is allowed there by NIR schema, thus xmLegesMarker sometimes has to deal with very complex cases (Fig. 1).

3.5 Tackling syntax errors

One of the main problems that have to be faced working with legacy documents and, generally, with documents edited manually, with no drafting support, is represented by the presence of syntactical errors. They can be:

- numbering errors;
- incorrect use of punctuation marks;
- errors in the layout of the document;
- unbalanced quotes;
- other drafting errors.

Some of these errors in the plain document have a limited effect in the XML output of xmLegesMarker, while others may cause totally disruptive behaviors of the automata used for the body parsing. For example, if quotes aren't balanced, the automata jams in the amendment states, forcing all the remaining text into the amendments tags. Another example, with not such a catastrophic effect, is represented by the erroneous or missing punctuation marks that should be used to separate different paragraphs inside an article; in that case, the next paragraph isn't acknowledged because of the erroneous separator, and all the remaining paragraphs in the article aren't acknowledged too, because of the checks on paragraph numbering, until the next article is read, which force a reset of the paragraph counter.

Thus, little errors in the input source often generate huge troubles in the resulting XML, while a little correction of the input saves the user from a painful correction of the output in an XML editor. For this reason a messaging system which allows the user to operate directly in the source, correct it and process it again was implemented inside xmLegesMarker.

The new version of the marker is able to identify the most typical troublesome situations having reference to errors in the source, embedding a warning message in the output. The message is formatted as a processing instruction in the resulting XML, so it doesn't affect the validity of the document. Moreover, the message contains a warning code that refers to a warning table where typical troubles are classified, described and a guideline to solve each of them is provided.

4 Qualitative evaluation

The main automata, dedicated to the analysis of the enacting terms section (the body of the document), consist of 83 regular expressions, 22 states (or start conditions) and more than 70 rules based on start conditions and stacks of start conditions. It handles ten different kind of partitions (book, part, chapter, title, section, article, paragraph, alphabetical, numerical and bulleted list), and various other entities defined in the NIR schema, like partition title, decoration, amendment, and so on. The lex file that defines the Flex scanner comprises more than one thousand lines of code.

4.1 Shape of legislative acts

Legislative documents may be particularly complex from a structural point of view. Let's have a look at a couple of numerical example.

In Italy, the bigger legislative documents are probably the Budget laws. The bill 1183[6] of 2007, for example, representing the 2007 Budget bill, counts in the main act, including the amendments, 1122 paragraphs, 46 articles, 410 alphabetical list partitions and 75 numerical list partitions, altogether 1653 partitions!

The nesting of the body is variable as well, but isn't uncommon to run into bills with almost ten nested partitions. For example, the bill 3328[7] of 2005, has, take a deep breath, a four-points-long numerical list inside the letter "a" in paragraph "3" of article "165-ter" within section "VI-bis" inside an amendment in paragraph "1" within article "6" of chapter "III" in the title "I"!

4.2 On-road test

The Italian Senate provided a big data-set of bills on which several tests have been carried out, aiming at more and more refining the software. Given the fact that is a pretty hard task to conduct a precise statistical analysis about the accuracy on the whole data-set, because of the complexity of the input, in this section we try to give at least a qualitative idea about the capabilities of xmLegesMarker, just reporting the marking-up process outcomes of the two, quite representative, previously discussed bills.

Table 1. Bill 3328 mark-up details

Partition	Total	Amendment	Missing
Title	6	0	-
Chapter	10	1	-
Section	4	4	-
Article	81	39	-
Paragraph	249	155	-
Alphabetical	178	73	-
Numerical	50	8	-

[6] http://www.senato.it/leg/15/BGT/Schede/Ddliter/27212.htm

[7] http://www.senato.it/leg/14/BGT/Schede/Ddliter/22640.htm

Table 2. Bill 1183 mark-up details, before and after correction of the original input

	Before correction			After correction		
Partition	Total	Amendment	Missing	Total	Amendment	Missing
Article	35	17	11	46	28	-
Paragraph	1072	867	50	1122	152	-
Alphabetical	398	324	12	410	98	-
Numerical	66	59	9	66	28	9

Although quite complex, the mark-up of 3328 body was perfect and the marker didn't miss any partition. On the other hand, the same process on the huge 1183 triggered two warning messages:

- erroneous balancing of quotes in art. 18 paragraph 45;
- not numbered comma inside an amendment in art. 18.

As discussed, the first one is a disruptive problem, which effectively causes a pretty bad result. After correcting the two problems in the original plain input, the marking-up process was performed again and the outcome was excellent: the only imperfection is given by two non-standard numerical lists ("1.1", "1.2", etc.), a format not included in the rules for legislative drafting enacted by the Parliament and, consequently, neither included in the NIR drafting standards.

Tables 1 and 2 report, for each type of partition, the number of total occurrences found by the marker, how many of them are found in amendments and how many are missing.

5 An Italian Senate application

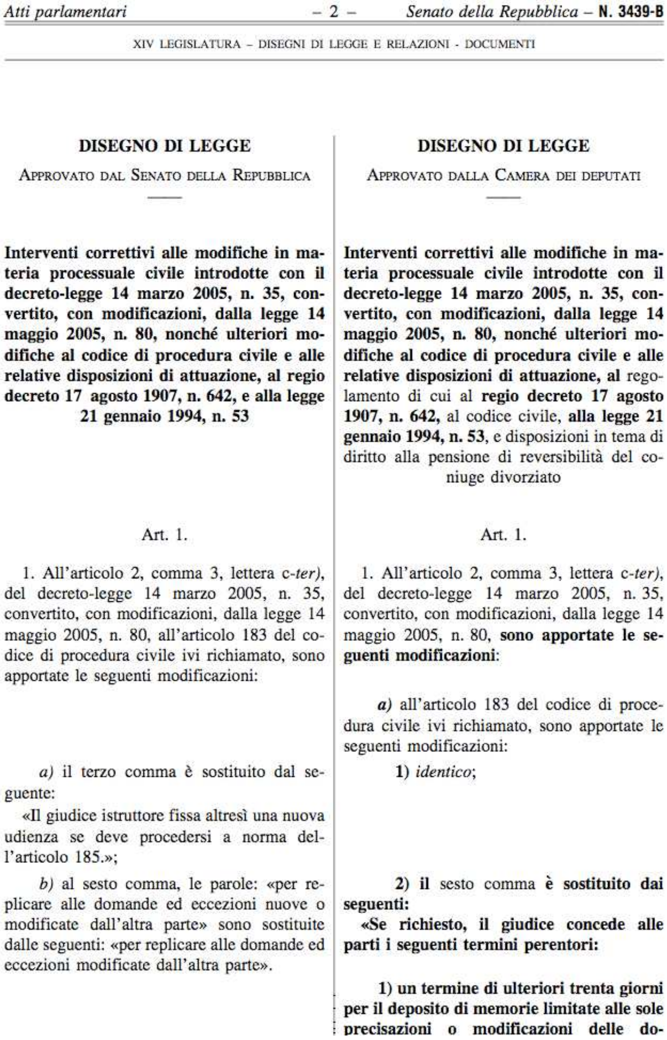
This section shows how the promising results of xmLegeMarker are exploited for evaluations within the TafWeb application, supported by the Italian Senate.

5.1 Scenario

The article by article discussion of a newly proposed bill is scheduled within the so-called "ordinary legislative procedure", in one of the two chambers of the Italian Parliament. During the discussion, amendments are voted and applied to the bill. Once the agreement is reached, the amended bill moves to the other Chamber, which is entitled to apply further modifications and send it back to the previous Chamber, until the bill is applied in a Chamber without introducing new modifications, which terminates the process.

During the process, the effects of approved amendments, i.e., the differences between the two versions of a bill under debate, are represented using a TAF document, which stands for Testo A Fronte, a parallel text organized in two columns, with the original text on the left and the modified text on the right (Fig. 2). The TAF document is very useful for two reasons:

- it highlights the effects of amendments by using specific textual representations for each kind of modification, making it easier to understand where and how a bill has been modified;
- it can be used to limit the analysis and the discussion of the bill only to the changed parts.



Atti parlamentari – 2 – *Senato della Repubblica* – **N. 3439-B**

XIV LEGISLATURA – DISEGNI DI LEGGE E RELAZIONI - DOCUMENTI

DISEGNO DI LEGGE Approvato dal Senato della Repubblica	**DISEGNO DI LEGGE** Approvato dalla Camera dei deputati
Interventi correttivi alle modifiche in materia processuale civile introdotte con il decreto-legge 14 marzo 2005, n. 35, convertito, con modificazioni, dalla legge 14 maggio 2005, n. 80, nonché ulteriori modifiche al codice di procedura civile e alle relative disposizioni di attuazione, al regio decreto 17 agosto 1907, n. 642, e alla legge 21 gennaio 1994, n. 53	**Interventi correttivi alle modifiche in materia processuale civile introdotte con il decreto-legge 14 marzo 2005, n. 35, convertito, con modificazioni, dalla legge 14 maggio 2005, n. 80, nonché ulteriori modifiche al codice di procedura civile e alle relative disposizioni di attuazione, al** regolamento di cui al **regio decreto 17 agosto 1907, n. 642,** al codice civile, **alla legge 21 gennaio 1994, n. 53**, e disposizioni in tema di diritto alla pensione di reversibilità del coniuge divorziato
Art. 1.	Art. 1.
1. All'articolo 2, comma 3, lettera c-*ter)*, del decreto-legge 14 marzo 2005, n. 35, convertito, con modificazioni, dalla legge 14 maggio 2005, n. 80, all'articolo 183 del codice di procedura civile ivi richiamato, sono apportate le seguenti modificazioni:	1. All'articolo 2, comma 3, lettera c-*ter)*, del decreto-legge 14 marzo 2005, n. 35, convertito, con modificazioni, dalla legge 14 maggio 2005, n. 80, **sono apportate le seguenti modificazioni**:
	a) all'articolo 183 del codice di procedura civile ivi richiamato, sono apportate le seguenti modificazioni:
a) il terzo comma è sostituito dal seguente: «Il giudice istruttore fissa altresì una nuova udienza se deve procedersi a norma dell'articolo 185.»;	**1)** *identico*;
b) al sesto comma, le parole: «per replicare alle domande ed eccezioni nuove o modificate dall'altra parte» sono sostituite dalle seguenti: «per replicare alle domande ed eccezioni modificate dall'altra parte».	**2) il** sesto comma **è sostituito dai seguenti:** **«Se richiesto, il giudice concede alle parti i seguenti termini perentori:**
	1) un termine di ulteriori trenta giorni per il deposito di memorie limitate alle sole precisazioni o modificazioni delle do-

Fig. 2. An excerpt of a TAF document

5.2 TafWeb

TafWeb[8] is an experimental web service developed by the Department of Computer Science of the University of Bologna under the supervision of the Italian Senate, which aims at automatically spawning the first version of a TAF document, in order to reduce the amount of work done for producing these documents from scratch. The overall system is currently under development for evaluation purposes. The core of TafWeb is represented by JNdiff[9], arisen from Ndiff [7][8], a highly configurable algorithm for smartly comparing XML documents.

Thanks to the integration of xmLegesMarker inside the TafWeb environment, it's possible to implement a service which is able, starting from the plain Chamber and Senate version of a bill, to automatically produce a document representing the TAF. The main steps involved are:

- the conversion of the plain Senate and Chamber version of a bill in XML through xmLegesMarker;
- the computation of the "difference document", through JNdiff;
- the application of a style sheet to the original and to the difference document, generating the TAF in the official formats: XHTML, Office Open XML, PDF.

6 Conclusions

In this paper we presented xmLegesMarker, a powerful parser for Italian legislative documents. Its main features and capabilities were deeply analyzed and a qualitative evaluation concerning the marking-up accuracy on plain bills was provided.

Besides the benefits in the information retrieval field, the conversion of a legislative corpus into the XML language permits the development of useful, legislative domain related software. The scientific collaboration between Italian Senate and Ittig, currently focusing on the promising outcomes of the marker, enabled the realization of a process to automatically produce the first version of a parallel text from the plain Chamber and Senate versions of a bill under debate. Within the TafWeb environment, JNDiff, a comparison algorithm for XML documents, exploits the very detailed mark-up generated by xmLegesMarker in order to capture amendments and structural differences.

The automatic production of the parallel text, a process manually performed to date, reduces the burden of communication between the two chambers of the Italian Parliament and speeds up the legislative amending process.

7 Acknowledgements

This work is supported by a grant of the Office for the development of the information systems of the Italian Senate.

[8] http://sourceforge.net/projects/tafweb/

[9] http://sourceforge.net/projects/jndiff/

References

1. Biagioli, C., Francesconi, E., Spinosa, P., Taddei, M.: The NIR project: Standards and tools for legislative drafting and legal document web publication. In Proceedings of ICAIL Workshop on e-Government: Modeling Norms and Concepts as Key Issues, pp. 69-78 (2003).
2. Agnoloni, T., Francesconi, E., Spinosa, P.: xmLegesEditor: an opensource visual XML editor for supporting legal national standards. In Proceedings of the V Legislative XML Workshop, European Press Academic Publishing, pp. 239-251 (2007).
3. Biagioli, C.: Ipotesi di modello descrittivo del testo legislativo per l'accesso in rete a informazioni giuridiche. Informatica e Diritto 2:90 (2000).
4. Spinosa, P.: Identication of legal documents through URNs (uniform resource names). In Proceedings of the EuroWeb 2001, The Web in Public Administration (2001).
5. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE 77 (2): pp. 81-106 (1989).
6. Francesconi, E.: The "Norme in Rete"- project: Standards and tools for Italian legislation. International Journal of Legal Information, vol. 34, no. 2, pp. 358-376 (2006);
7. Schirinzi, M., Vitali, F., Di Iorio, A.: Ndiff, un approccio naturale al confronto di documenti XML (2007).
8. Di Iorio, A., Marchetti, C., Schirinzi, M., Vitali, F.: A Natural and Multi-layered Approach to Detect Changes in Tree-Based Textual Documents. (to appear) Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS), (2009).