NLP–based Metadata Extraction for Legal Text Consolidation

PierLuigi Spinosa
Institute of Legal Information Theory and Techniques
via de' Barucci, 20
Firenze, Italy
p.spinosa@ittig.cnr.it

Gerardo Giardiello
Institute of Legal Information Theory and Techniques
via de' Barucci, 20
Firenze, Italy
g.giardiello@ittig.cnr.it

Manola Cherubini
Institute of Legal Information Theory and Techniques
via de' Barucci, 20
Firenze, Italy
m.cherubini@ittig.cnr.it

Simone Marchi
Institute of Computational Linguistics
via G. Moruzzi, 1
Pisa, Italy
simone.marchi@ilc.cnr.it

Giulia Venturi
Institute of Computational Linguistics
via G. Moruzzi, 1
Pisa, Italy
giulia.venturi@ilc.cnr.it

Simonetta Montemagni
Institute of Computational Linguistics
via G. Moruzzi, 1
Pisa, Italy
simonetta.montemagni@ilc.cnr.it

ABSTRACT

The paper describes a system for the automatic consolidation of Italian legislative texts to be used as a support of an editorial consolidating activity and dealing with the following typology of textual amendments: repeal, substitution and integration. The focus of the paper is on the semantic analysis of the textual amendment provisions and the formalized representation of the amendments in terms of metadata. The proposed approach to consolidation is metadata–oriented and based on Natural Language Processing (NLP) techniques: we use XML–based standards for metadata annotation of legislative acts and a flexible NLP architecture for extracting metadata from parsed texts. An evaluation of achieved results is also provided.

Keywords

Natural Language Processing, textual amendments, XML representation, metadata extraction, consolidation of legal text

1. INTRODUCTION

The consolidation of the legislative act during its life cycle represents a central research area in the AI and Law field. If on the one hand fully automatic consolidation still appears a challenging task, on the other hand it is widely known that consolidation can be carried out semi–automatically with the help of automatic procedures supporting different steps of the consolidation process. As a matter of fact, recent information technologies increasingly helped to facilitate the creation and updating of consolidated versions of the legislation currently in force. According to the recently published Interim Report of the Working Group on "Consolidation" of the European Forum of Official Gazettes [1], it appears that in Belgium and Germany some steps of the consolidation process are supported by automatic procedures, although the modification of the text is manually done; in Slovakia dedicated software is being developed to carry out consolidation in a fully automatic way. In Japan, an automatic consolidation system for Japanese statutes has been developed based on the formalization and experts' knowledge about consolidation [2].

This work deals with automatic consolidation of legislative texts as a support of an editorial consolidating activity and copes with the following textual amendments: repeal, substitution and integration. The consolidation process consists in the integration within a single text of the provisions of the original act together with all subsequent amendments to it. Different issues are at work here from the formal XML representation of legislative acts to accessing the content of legislative acts taken as input which requires understanding the linguistic structures of the text. Although legal language is much more constrained than ordinary language, nevertheless its syntactic and lexical structures still pose a considerable challenge for state–of–the–art linguistic technologies. In our view, the general consolidation workflow (see Figure 1) can be seen as organized into the following steps:

1. semantic analysis of the textual amendment provisions;
2. formalized representation of the amendments by a metadata set;
3. proper text modifications performed on the basis of the metadata interpretation and
4. production of the consolidated text.

In this paper, we report the results of a joint research effort carried out by ITTIG-CNR (Institute of Legal Information Theory and Techniques of the Italian National Research Council) in collaboration with ILC–CNR (Computational Linguistics Institute) aimed at developing a semi-

ICAIL-2009 Barcelona, Spain

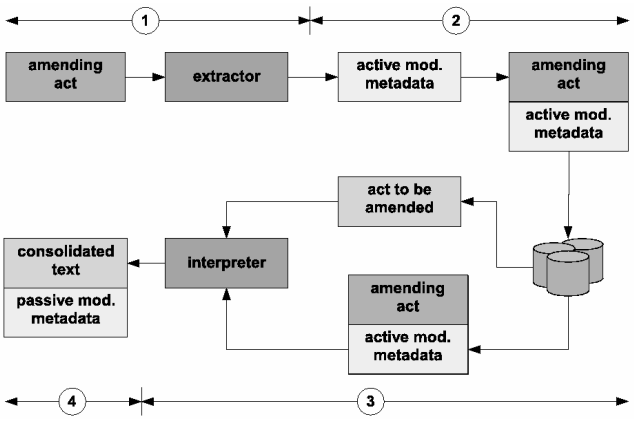

Figure 1: The consolidation workflow

automatic consolidation system to propose a candidate consolidated text which has to be validated by the editorial staff in charge of the process who can accept, correct or refuse the automatically generated text. In particular, in this paper we will focus on steps 1. and 2. of the consolidation work flow (namely semantic analysis and formalised representation of the provision text), which are the most critical ones of the whole work flow.

The proposed approach to consolidation is metadata–oriented and based on Natural Language Processing (NLP) techniques. Our strategy makes use of i) XML–based standards for metadata annotation of legislative acts, and ii) a flexible NLP architecture for extracting metadata from parsed texts. In what follows, after a short description of the results of a preliminary analysis and classification of textual amendments in Italian legislative acts (section 2), the metadata scheme adopted for the representation of consolidated texts and the metadata extraction process are described in detail respectively in sections 3 and 4, while the evaluation of the extraction system is presented in section 5. In section 6 current directions of research and development are presented; finally some conclusions are reported (section 7).

2. ANALYSIS AND CLASSIFICATION OF TEXTUAL AMENDMENTS

As a preliminary step, an accurate study has been carried out on the typology of textual amendment provisions which are present in Italian legislative texts, involving both structure and word modifications. To this specific end, a representative sample of about 800 textual amendment provisions has been collected, where each provision has been classified at three different levels:

1. amendment type, namely repeal, integration or substitution;
2. typology of modified objects, e.g. partitions, periods or words;
3. sub–type of the specific partitions (article, paragraph, etc.) or linguistic subdivisions (period, "alinea", etc.) involved in the amendment.

All the collected examples have been classified with respect to the parameters listed above, with particular attention to the variety of linguistic expressions used by the legislator to convey the different amendment types.

3. METADATA SCHEME

The national standard defined by the NormeInRete (NIR) project[1] has been considered as a starting point, since it provides specifications for describing the functional aspects of a norm; in this context a legislative text is considered as a set of provisions rather than partitions. In case of provisions concerning text modification, a set of metadata has been defined describing the norm to be modified (with a unique identifier), the text to be amended ("novellando"), the text to be inserted ("novella"), etc. From the analysis of the collected sample of textual amendment provisions, the NormeInRete standard turned out to be not sufficient to describe adequately the modifying action to be performed.

Before going into details, a number of essential features of the NormeInRete national standard have to be highlighted:

1. it aims to describe all properties of a document in an independent and exhaustive way; in the case of a modifying act, it should express the activity without accessing or intervening on the act under modification;
2. it provides specific rules to construct IDs of the XML elements;
3. it does not deal with either the period as an XML element or mandatory ID for some partitions (e.g. partition title) and for linguistic subdivisions ("alinea", closing paragraph, etc.).

Different from other countries, in the Italian legal system modifications not only concern formal partitions, but also linguistic sub–parts up to single words, so they may contain references to objects which have no mandatory ID or are not elements.

Furthermore, in many cases the ID value, even for partitions where it is mandatory, cannot be predicted, because some hierarchical levels in the citations can be missing (see for instance "letter a) of art. 5", where no explicit paragraph indication is provided), and also because the relative ordering of partitions (last, second–last, etc.) is often used. On the basis of these considerations, an extension of the NIR national scheme has been designed, formalized in a proposal which was submitted to the national XML standards Working Group and which is currently implemented in terms of proprietary metadata. The proposal is based on the following principles:

1. the area delimiting a modification is, first of all, described by a set of nested containers;
2. the hierarchy of containers goes from formal partitions to linguistic sub–parts;
3. each container can be identified by a unique label (e.g. *point c.*) or a position in the list, expressed by an ordinal number (e.g. *second period*) or through its relative order (e.g. *last paragraph*);
4. within the narrowest container, the exact point can be indicated through absolute positions (start, end) or relative positions (after, before, etc.) with respect to existing words within quotes;

[1] http://www.normeinrete.it

5. consequently the container where the modification occurs is indicated firstly by the narrowest partition whose ID can be calculated; other elements can be possibly nested in this element, each of which is identified in terms of a given type (e.g. paragraph, "alinea", period, etc.) and a label or a position.

In particular, the extensions are concerned with the following cases:

- the insertion of a set of elements, the so–called "border", that can be recursively nested, for representing narrower containers of a norm, since it can point at a single partition ID. Such new element has the attributes related to a container type and a label or position;
- the explicit mention of the type of "novella" or "novellando" that is the entity (partition, linguistic sub–part or words) involved in the modification.

The extended metadata set describing a textual modifying provision is reported in Table 1, where the hierarchical level is indicated with dash ("–") indentation and attributes are separated by a colon (":") from the relative tag. Note that all metadata are empty elements, without any textual content, and the information is conveyed in terms of attributes.

With such an extension the modification is completely described and the following functions can be automatically carried out:

1. isolating the narrowest container, even when the ID is missing, by functionalities able to identify a period, searching for an element with a given label, or calculating its absolute or relative position;
2. positioning at the boundaries or within the container, through searching a given string of characters.

Therefore, through automatic interpretation of these metadata, it is possible to perform the above mentioned modifications on the text and to submit a proposal for an updated text for the editorial activity.

Once the metadata scheme for the representation of textual amendments has been defined, it was applied to the representative sample of textual amendments collected during the analysis stage. The final result is a manually annotated corpus of textual amendments (henceforth referred to as "gold standard"), to be used as a reference corpus for the development of the metadata extraction system (see below).

4. METADATA EXTRACTION

Metadata extraction is performed by the MELT ("Metadata Extraction from Legal Texts") system, whose overall architecture is depicted in Figure 2.

MELT has a three–module architecture composed by:

1. the *xmLeges tools* in charge of preparing the text for further processing stages, by a) identifying normative references, b) analyzing the formal structure of the normative act, and c) classifying individual provisions into coarse–grained classes. This classification is aimed at identifying amendment provisions for them to be selected and passed to further processing stages;

Metadata	Description
pos	information on the amending provision
pos:xlink	ID reference to the amending provision
norm	information on the norm to be amended
norm:xlink	URN reference to the norm
– pos	further information on the norm
– pos:xlink	URN reference to the norm with the partition ID
– – border	information on further narrower container
– – border:type	container type (e.g. point, "alinea", period, etc.)
– – border:num	container label expressed by a number or a letter
– – border:ord	container position expressed by an ordinal (e.g. 2nd) or a relative (e.g. last) number
position	information on the specific modifying point within the narrowest container
– pos	information on a string (quoted) and a bound of the deleting or inserting point
– pos:xlink	ID reference to the string, a bound of which is the beginning of the modifying text
– pos:where	specific bound of the string or container (before, after, start, end)
novellando	information on the outgoing text
– type	information on the "novellando" type
– type:value	"novellando" type (e.g. article, paragraph, "alinea", period, words, etc.)
– pos	information on the outgoing string (in quotes)
– pos:xlink	ID reference to the string that is either the outgoing text, or the beginning or ending of the outgoing text
– – role	information on the meaning of the string
– – role:value	string role: beginning (from) or ending (up to) of the outgoing text
novella	information on the incoming text
– type	information on the "novella" type
– type:value	novella type (e.g. article, paragraph, "alinea", period, words, etc.)
– pos	information on the incoming string (quoted)
– pos:xlink	ID reference to the incoming string

Table 1: Metadata set common to each textual amendment provision

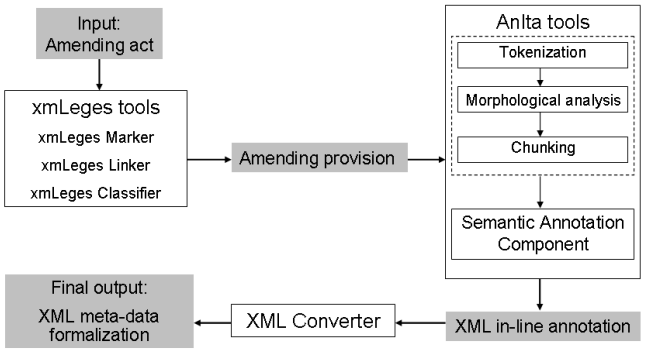

Figure 2: MELT overall architecture

2. the *AnIta tools*, a suite of Natural Language Processing tools for the analysis of Italian texts, in charge of a) the linguistic analysis of the selected provisions (going from tokenization and morphological analysis to shallow syntactic parsing), and b) their semantic annotation with the typology of semantic tags reported in Table 1;

3. an *xml Converter*, in charge of the conversion of the in–line annotations produced by the AnIta tools into the final XML metadata representation format described in Section 3.

The system can be seen as a development of SALEM (Semantic Annotation for LEgal Management[3]), an NLP–based system developed for automatically producing a semantic annotation of Italian legal texts (including modification provisions) which was used as an advanced module of the NIREditor (now xmLegesEditor) [4] to support the legal drafting process. The original system has been personalized in a number of different aspects, mainly related with a) the input type the system has to deal with, and b) range and typology of metadata to be extracted. Namely, while legal text analysis in the SALEM framework is driven by a broad *ontology* of legislative provision types (including, among others, obligations, permissions, prohibitions, etc.), the current system has been intended to focus on *amendment provisions* only. On the other hand, the typology of metadata to be extracted is much wider in this case, being instrumental to the semi–automatic generation of consolidated versions of the legislation currently in force.

4.1 Pre–processing of the amending act with xmLeges tools

As Figure 2 shows, the first step of the metadata extraction process is performed by the *xmLeges tools* [5] operating through the following stages: transforming the raw normative text into an XML file where the formal structure of the text is made explicit (xmLegesMarker); detecting normative references (xmLegesLinker); classifying provisions into macro–classes (xmLegesClassifier). Since in the present case we are dealing with amendment provisions only, all other provision types recognized at this stage are discarded from further processing.

This pre–processing step is also in charge of preparing the text for the linguistic analysis by packing the content of normative references and quoted text and by replacing it with placeholders to be used subsequently to recover the original text. For instance, the repeal provision

> "All'articolo 1, comma 1, della legge 8 febbraio 2001, n. 12, la lettera d) é abrogata" (*In article 1, paragraph 1, of the act 8 February 2001, n. 12, letter d) is repealed*)

is transformed into

> "All' REF mod31-rif2#art1-com1, la lettera d) é abrogata" (*In REF mod31-rif2#art1-com1, letter d) is repealed*)

where it can be noticed that the provision text has been noticeably simplified without information loss. The relevance of this choice is motivated by the notorious complexity of legal language, characterised by the frequency of occurrence of deep chains including a high number of embedded prepositional chunks (typically corresponding to intra-textual and inter-textual cross–references) which make the legal text quite difficult to be processed (see [11] for details).

4.2 Semantic processing of the amending act

The text pre–processed by the *xmLeges tools* is then passed to the Natural Language Processing modules (hereafter, referred to as *AnIta tools*) [10] which carry out the text analysis in two different steps:

1. the input text is first parsed by the linguistic modules in the dashed box of Figure 2 providing as output a shallow syntatic analysis of the amendment provision;

2. the shallow parsed text is then fed into the *Semantic Analysis Component*, with the result of deriving and making it explicit the semantic content implicitly stored in the analysed provision.

4.2.1 Linguistic analysis

The linguistic analysis stage creates the data structures on which the further processing stage operates. During this step, the input text is first tokenized and normalized for dates, abbreviations and multi–word expressions; the normalized text is then morphologically analyzed and lemmatized, using an Italian lexicon specialized for the analysis of legal language; finally, the text is POS–tagged and shallow parsed into non–recursive syntactic constituents called "chunks".

In the architecture of MELT, text "chunking" plays a central role, representing the very starting point of metadata extraction in its own right. It is carried out through a battery of finite state automata (CHUG–IT [6]), which takes as input a morphologically analysed and lemmatised text and segments it into an unstructured (non–recursive) sequence of syntactically organized text units, the so–called "chunks". A chunk is a textual unit of adjacent word tokens sharing the property of being related through dependency relations (es. pre–modifier, auxiliary, determiner, etc.). A sample output of this syntactic processing stage is given in Figure 3, where the input sentence is segmented into eight chunks. It can be noted that each chunk contains information about its type (e.g. prepositional chunk or P_C, nominal chunk or N_C, finite verbal chunk or FV_C, punctuation chunk or PUNC_C), its lexical head (identified by the label *potgov*) and any occuring determiner, auxiliary verb or preposition.

All' REF MOD31-RIF1#ART1-COM1, la lettera d) é abrogata.
'In REF MOD31-RIF1#ART1-COM1, letter d) is repealed'
[[CC:P_C][PREP:A#E][DET:LO#RD@FS@MS][POTGOV:REF#SP@NN]]
[[CC:U_C][FORM:MOD31-RIF1#ART1-COM1]]
[[CC:PUNC_C][PUNCTYPE: ,#@]]
[[CC:N_C][DET:LO#RD@FS][AGR:@FS][POTGOV:LETTERA#S@FS]]
[[CC:N_C][AGR:@FP@FS@MP@MS][POTGOV:D#S@FP@FS@MP@MS]]
[[CC:PUNC_C][PUNCTYPE:)#@]]
[[CC:FV_C][AUX:ESSERE#V@S3IP][POTGOV:ABROGARE#V@FSPR]]
[[CC:PUNC_C][PUNCTYPE: .#@]]

Figure 3: A sample of chunked text

It should be noticed that at this stage no information about the nature and scope of inter–chunk dependencies has been provided yet.

Although it might seem that full parsing should in principle be preferred for carrying out a metadata extraction task, we can think of a number of reasons for exploiting *chunking* as a starting point for metadata extraction from legal texts. Rather than producing a complete analysis of sentences, chunking representation provides only partial analysis allowing robustness and flexibility of the system in face of parse failures. The resulting analysis is flat and unambiguous: only those relations which can be identified with certainty have been found out and explicitly marked–up. Accordingly, structural ambiguities specific of legal language (but not only), such as prepositional phrase attachments, or long–range dependency chains, etc. are left underspecified and unresolved. Even if quite rudimentary, this first level of syntactic grouping is successfully instrumental for the identification of deeper levels of linguistic analysis.

Interestingly to our scope, chunking makes also available information about low level textual features as well as about the linear order of chunks. For example, information about *punctuation*, which is typically lost at further levels of analysis or – when it is present – it is hidden in the syntactic structure of the text, plays a quite crucial role in the identification of textual subparts within amendment provisions (such as "borders" of the norm to be modified, the exact position of amending parts, etc.). As it can be seen in the chunked sentence reported in Figure 3, punctuation chunk types (PUNC_C) are still part of this shallow syntactic representation, so that their presence and order with respect to other chunk types can be tested in the definition of the specialized set of syntactic rules for metadata extraction (see below).

4.2.2 Semantic analysis

As Figure 2 illustrates, chunked representations are fed into the *Semantic Annotation Component*, a finite–state compiler of grammars using a specialized grammar aimed at extracting the metadata relevant for the description of the three selected amendment sub–classes, i.e. repeal, integration and substitution. Rules in the grammar have the following generic form:

<chunk-based regular expression WITH set of tests => actions>

The recognition of the amending act type as well as of "structural" information concerning the subpart of the norm being modified, or of the "novella" and/or "novellando" (see Table 1) is carried out on the basis of patterns formalized in terms of regular expressions operating over sequences of chunks. These patterns include both lexical and syntactic conditions which are checked through a battery of tests; the latter can also include checks aimed at identifying basic syntactic dependencies (e.g. subject, object) amongst chunks. The action type of such rules consists in the extraction of relevant metadata from the linguistically pre–processed text.

The classification of modification provision is based on a combination of both syntactic and lexical criteria. As expected, a strong association has been observed between the verbs used to convey this information and the amendment type. For instance, a "repeal" provision is typically expressed by verbs such as *abrogare*, *sopprimere*, *eliminare* (respectively, "to repeal", "to delete", "to remove"), an "integration" provision by verbs such as *aggiungere*, *inserire* ("to add", "to insert"). Despite the lexical regularities observed with respect to the provision type, the syntactic structure of provisions is not so easily predictable and poses a considerable challenge in the development of the grammar; the definition of structural patterns over sequences of chunks had to cope with the observed variability of the syntactic structures underlying amending sentences.

To keep with the example in Figure 3, the following metadata analysis has been provided. First, the modification provision has been classified as a "repeal", resulting in the following in–line metadata annotation:

```
All' <norm>REF mod31-rif1#art1-com1#</norm>,
la <border>
<border:type>lettera</border:type>
<border:num>d)</border:num>
</border> e' abrogata.
```

where the value of `norm` is the placeholder of the amending act generated during the pre–processing stage by the xmLeges tools, and where the values of `border:type` and `border:num` jointly identify the specific partitions of the norm being modified. In this specific case, the `novellando` (i.e. the text to be amended) coincides with the `border` of the norm to be modified.

The final metadata extraction step is performed by the *xml Converter*, a component in charge of converting the in–line annotations produced by the *AnIta tools* into a metadata description conformant to the XML metadata representation specifications detailed in Section 3.

4.2.3 Related work

The approach to metadata extraction described in this paper is closely related to the task of Information Extraction as defined in the literature. Information Extraction (IE) is the task of identifying, collecting and normalizing relevant information from natural language texts while skipping irrelevant text passages; IE systems thus do not attempt to offer a deep, exhaustive linguistic analysis of all aspects of a text; rather, they are designed to "understand" only those text passages that contain information relevant for the task at hand.

In our system, we perform the mapping from natural language sentences in the amendment provisions to corresponding domain knowledge: in particular, the resulting mappings turn free text into target knowledge structures, containing crucial information such as the norm being modified, the amending and the amended text, etc. Target knowledge structures are not arbitrary, but rather predefined by an

ontology providing a formal specification of a shared understanding of the domain of interest (see Section 3). Operationally, our system relies on document pre–processing and extraction rules to identify and interpret the information to be extracted.

[7] reports similar work concerned with the automatic semantic interpretation of legal modificatory provisions: based on a taxonomy of modificatory provisions compliant to the NormeInRete standard, a system for the identification of semantic frames associated with a modification type has been developed. The developed system shares a number of features with our approach. In both cases, NLP technologies are resorted to: semantic analysis is carried out on the linguistically (syntactically) pre–processed text on the basis of a rule–based approach. The main difference is concerned with the starting point of the semantic analysis stage: [7] relies on a *deep syntactic analysis* of the provision text, whereas our system starts from the *shallow syntactically parsed text*, whose output is *underspecified*. [7] motivates this choice on the basis of the fact that a range of complex syntactic phenomena (e.g. coordinated structures and relative clauses) that cannot be properly accounted for in a *shallow parsing* approach can be covered if *deep parsing* is resorted to. Obviously, we can only agree with such a motivation. However, on our view of things, there are two main reasons for opting for a different approach. First, in the shallow parsed text information about the linear order of identified chunks is still available, which makes it possible to enforce linear precedence constraints which otherwise would be very difficult to be expressed: these constraints are very useful to test for the presence of low level textual features such as punctuation. Second, and most importantly here, currently the performance of state–of–the–art chunkers and dependency parsers differs significantly: for chunking the F–score of state–of–the–art systems ranges between 91.5 and 92.5 (with the precision score going from 88.82% to 94.29%, see [12]), whereas the performance of dependency parsers is much less reliable. In the CoNLL 2007 Shared Task on Dependency Parsing [13], the average Labelled Attachment Score (LAS) over all systems varies from 68.07 for Basque to 80.95 for English, with top scores varying from 76.31 for Greek to 89.61 for English.

Given the encouraging results we achieved with our system so far (see Section 5), we believe that for metadata extraction from the amending act the shallow parsed text is a sufficient and even preferable starting point. In particular, in our approach a better balance between accuracy and robustness can in principle be achieved; it goes without saying that this statement needs careful evaluation. However, this is in line with [8] who claim that in many natural language applications it is sufficient to use shallow parsing: this information type has been found successful in tasks such as information extraction and text summarisation (concerning the legal knowledge management, see among others [9]).

4.3 An example

In this section, each step of the metadata extraction workflow is exemplified. As Figure 1 illustrates, the starting point is an amending act whose amendment provisions (such as the one reported below) are subject to further processing stages.

Original input:

Nell'articolo 13–bis del testo unico delle imposte sui redditi, approvato con decreto del Presidente della Repubblica 22 dicembre 1986, n. 917, e successive modificazioni, concernente detrazioni per oneri, al comma 1, lettera c), secondo periodo, dopo le parole: "dalle spese mediche" sono inserite le seguenti: "e di assistenza specifica (*In the article 13-bis of the consolidated text on the income taxes, approved by the President of Republic decree 22 December 1986, n. 917, and subsequent modifications, relating to tax allowances, in paragraph 1, point c), second period, after the words: "medical costs" the following ones are inserted: "and of treatment".*)

The *xmLeges tools* are in charge of pre–processing the raw text (as described in Section 4.1) and provide as output the XML marked–up text exemplified below:

xmLeges tools output:

```
<mod id="mod1">
Nell' <ref xlink:href="urn:nir:stato:testo.unico;
imposte.redditi:1986-12-22;917#art13bis">articolo
13-bis del testo unico delle imposte sui
redditi</ref>, approvato con
<ref xlink:href="urn:nir:presidente.repubblica:
decreto:1986-12-22;917>decreto del Presidente della
Repubblica 22 dicembre 1986, n. 917</ref>, e
successive modificazioni, concernente detrazioni
per oneri, al comma 1, lettera c), secondo periodo,
dopo le parole:
"<quotes id="mod1-vir1" type="words">dalle spese
mediche</quotes>" sono inserite
le seguenti: "<quotes id="mod1-vir2" type="words">
e di assistenza specifica</quotes>".
</mod>
```

In turn, the XML marked–up text is tranformed as follows, for it to be processed by the *AnIta tools* in charge of the semantic analysis stage:

Input of the AnIta tools:

```
Nell'REF MOD1-RIF1#art13bis, approvato con
REF MOD1-RIF2, e successive modificazioni,
concernente detrazioni per oneri, al comma
1, lettera c), secondo periodo, dopo le
parole: "QTS MOD1-VIR1" sono
inserite le seguenti: "QTS MOD1-VIR2".
```

As reported in Section 4.2.2, the metadata annotation of the pre–processed amending provision operates on the chunked representation of the input text. The output of the *Semantic Annotation Component* is exemplified below:

XML in–line metadata annotation:

```
<integration>
<norm>Nell'REF MOD1-RIF1_ART13BIS</norm>,
approvato con REF MOD1-RIF2, e successive
modificazioni, concernente detrazioni per oneri,
<border>al comma 1</border>, <border>lettera
c)</border>, <border>secondo periodo</border>,
<where>dopo</where>
<novella:type>le parole</novella:type>:
<position>"QTS MOD1-VIR1</position>" sono
```

```
inserite le seguenti: <novella>"QTS MOD1-VIR2
</novella>".
</integration>
```

The in–line metadata annotation exemplified above is then converted by the *xml Converter* into the XML final representation format described in Section 3, as exemplified below:

```
<dsp:integration>
   <dsp:pos xlink:href="#mod1" />
   <dsp:norm xlink:href="urn:nir:stato:testo.unico;
       imposte.redditi:1986-12-22;917">
     <dsp:pos xlink:href="urn:nir:stato:testo.unico;
         imposte.redditi:1986-12-22;917#art13bis"/>
      <dsp:subarg>
        <ittig:border type="comma" num="1" >
          <ittig:border type="lettera" num="c" >
             <ittig:border type="periodo" ord="2" >
               </ittig:border>
            </ittig:border>
          </ittig:border>
      </dsp:subarg>
   </dsp:norm>
   <dsp:position>
     <dsp:pos where="dopo" xlink:href="#mod1-vir1"/>
   </dsp:position>
   <dsp:novella>
      <dsp:pos xlink:href="#mod1-vir2" />
      <dsp:subarg>
         <ittig:type value="parole" />
      </dsp:subarg>
   </dsp:novella>
</dsp:integration>
```

Note that, independently from the order in which the different "border" values are expressed in the amendment provision text, within the resulting metadata description they are organised hierarchically from the broadest to the narrowest one.

5. EVALUATION OF RESULTS

The system has been evaluated on a sample of textual amendment provisions which were selected as representative of the typology of textual amendment subtypes and of the metadata to be extracted. This sample includes both structure and word modifications: it is representative of the three typologies of textual amendments considered (namely repeal, integration and substitution), with particular attention to the different typologies of modified objects (partitions, periods, words, etc.) and their linguistic expression within the text. The test corpus, constituted by 147 amendment provisions, was collected by law experts at ITTIG–CNR, who also provided a hand–annotated version of the selected sample. The aim of the evaluation was to assess the system's performance with respect to two different tasks: 1) classification of the amendment provisions into the subclasses of repeal, integration and substitution (henceforth referred to as "classification task"), and 2) metadata extraction (henceforth "metadata extraction task").

In both cases, evaluation is carried out in terms of Precision and Recall, where Precision is computed as the ratio of True Positives (or TP) over all system answers (including both TPs and False Positives or FPs), and Recall refers to the ratio of TPs over all provisions in the test corpus (corresponding to the sum of TPs with missed answers or False Negatives, FN).

Provision type	Total	TP	FP	FN	Prec	Recall
Repeal	62	62	0	0	1	1
Integration	49	49	0	0	1	1
Substitution	36	36	0	0	1	1

Table 2: Test corpus composition and classification results.

Provision type	Meta-data	TP	FP	FN	Prec	Recall
Repeal	297	285	4	12	0.986	0.960
Integration	361	344	2	17	0.994	0.953
Substitution	264	245	0	19	1	0.928
Total	922	874	6	48	0.993	0.948

Table 3: Metadata extraction results by provision subtype.

Table 2 reports the test corpus composition together with the results achieved for the classification task, where Precision is computed as the ratio of correctly classified provisions (TPs) over all system answers (corresponding to TPs+FPs), and Recall refers to the ratio of correctly classified provisions (TPs) over all provisions in the test corpus (corresponding to the sum of TPs with FNs). It can be noticed that the system reaches 1 for both Precision and Recall for all considered amendment provision subclasses, showing that the linguistic patterns used to convey this information type in the text are used unambiguously and can be resorted to reliably to carry out the classification task.

Tables 3 and 4 summarize the results of the metadata extraction task. The aim of the evaluation here was to assess the system's reliability in identifying, for each provision subtype, all the metadata that are relevant for that provision and are instantiated in the text. In particular, Table 3 records for each provision subclass the total number of metadata to be identified in the test corpus; this value was then compared with the number of metadata correctly identified by the system and the total number of answers given by the system. Here, Precision is scored as the number of correctly extracted metadata (TP) returned by MELT over the total number of returned metadata (TP+FP), while Recall is the ratio of correct metadata returned by the system (TP) over the number of expected answers (TP+FN).

Table 4 nicely complements the information contained in Table 3 by providing the same data organized differently, i.e. by metadata type. By comparing the results in the two tables, it can be noticed that no substantial differences in the metadata extraction performance are observed across the different provision subclasses dealt with (Table 3). On the other hand, results in Table 4 make it possible to identify the areas of the grammar which need further improvements. Concerning the latter, the lowest recall values are observed with respect to the "Position" metadata, in charge of indicating the exact point within the text where the modification should be performed. This appears to originate from the high linguistic variability through which "Position" informa-

Metadata class[:type]	Total	TP	FP	FN	Prec	Recall
Norm	146	144	2	2	0.986	0.986
Border :type	200	186	0	14	1	0.93
Border :num	193	177	3	16	0.983	0.917
Novellando :type	98	97	1	1	0.990	0.990
Novellando :pos	39	39	0	0	1	1
Novellando :role	1	1	0	0	1	1
Novella :type	84	81	0	3	1	0.964
Novella :pos	85	82	0	3	1	0.965
Position :where	52	46	0	6	1	0.885
Position :pos	24	21	0	3	1	0.875
Total	922	874	6	48	0.993	0.948

Table 4: Metadata extraction results by metadata type.

tion is expressed in legal texts; as a matter of facts, the different lexico–syntactic realizations of the positions where the text of the amendment (i.e. "Novella") had to be inserted made the definition of specific rules quite a challenging task.

It is interesting to note however that, on average, precision is higher than recall, ranging between 1 and 0.983; this means that the MELT system is significantly reliable in the answers it returns.

6. CURRENT DIRECTIONS OF RESEARCH AND DEVELOPMENT

The metadata extraction component has been integrated in the XMLeges-Editor[2] and is currently being tested at some Public Administrations: this is the first step of the process for assigning metadata to the textual modifications. The editorial activity is also supported by a facility for editing the metadata generated by automatic extraction, allowing their validation as well as possible integrations and corrections.

The whole consolidation process has been studied and designed and it is under construction; currently several components have been already developed. The consolidation workflow is completed by a specific metadata interpreter able to: a) access an amending act; b) localize, on the text to be amended, the portion under modification, and, c) perform a proper mark–up of deleted and/or inserted text. Here below such workflow is described.

After the insertion, and the possible correction, of extracted metadata into the proper section of the XML document, the amending act can be saved. In the designed workflow, this operation is performed by a CMS (Content Management System) in a centralized repository. The environment is powered by a native XML search engine (eXist), able to perform selections on the whole corpus exploiting completely the document mark-up. This search engine will be configured with specific queries, useful in the consolidation process: a) to retrieve all the acts to be amended and, for each of them, b) to extract all the active modification metadata to be applied. This retrieval process is deterministic since such queries exploit the XML structure of the active modification metadata which include a URN-based identifiers of the amending act.

The modifications metadata will be extracted and ordered according to the application date; at this point, the interpreter can be activated according to the following steps:

1. first of all it extracts the formal partition whose ID is specified in the metadata element "norm";
2. from this element, it extracts the first available child container (through the "border" tag) that satisfies both the given type as well as the label or the position; this step is repeated for all nested containers;
3. then, if necessary, it extracts the indicated period;
4. afterwards, the exact position, referred also to other parameters (start/end or words before/after), of the deleting or inserting operation is identified; in case of abrogation or substitution the outgoing text is identified as well;
5. and, finally, the possible outgoing and incoming text are marked–up and inserted in a multi–version format.

The result, in fact, is a multi–version document, that is a unique XML object that contains all the life cycle of the act. The container of each amended text portion has an attribute that indicates the in-force time interval.

Together with the generation of a consolidated text, the procedure will fill also the relative metadata set of the passive modification. As previously pointed out, in the Italian standard any document is autonomous and self-explaining. The following information (passive metadata, in large part derived by the active metadata set) will be reported:

- the type of the amendment (repeal, substitution or integration);
- the norm (via URN) and its partition (via ID) that caused the amendment;
- the "novellando" (text to be amended) and/or "novella" (amending text) components of the amendment, through links to their container IDs.

Moreover the life cycle of the act will be updated properly with event dates and passive relations.

As the metadata extractor, the interpreter will be integrated in the xmLeges-Editor which provides functionalities for verifying and correcting the proposed consolidated text and related metadata. The verification of the consolidated text correctness is facilitated by the application of specific style–sheets on the multi–version XML document. On the basis of the original text, any outgoing text (deleted or substituted) is shown in red while the incoming one is presented

[2]The XML Visual Editor developed by ITTIG provides a platform for drafting legal texts and produces documents in accordance with NormeInRete (NIR) Italian national standards.

in green and, for each of them, a punctual note with the in–force interval is inserted. Similarly the HTML document exportation for the browser is generated. Moreover it is possible to show passive metadata set related to each amended text portion and, through specific metadata forms and/or the normal editing functions a legal expert will be able to correct the consolidated text proposal.

7. CONCLUSIONS

The research activity described in this paper addresses the challenging task of automatic consolidation of the legislative act, whose final outcome is expected to be the proposal of a consolidated text, obtained through metadata interpretation, to be submitted to the editorial activity.

The tools developed are already being used with satisfying results in several editorial activities and in particular, in the production of in–force normative systems in some Italian Regions (Campania and Molise). The total time saving, in combination with the control and congruity of the inserted information, has been appreciated by the users. Even if a measure of the time and error saving has not been produced yet, current experiments carried out by the editorial staffs proved a qualitative effectiveness of the system, which is able to already reduce by more than 60% their whole efforts of the consolidation process related to textual amendments, covering a relevant number of the modification cases. This percentage can be increased by enlarging the coverage of the grammar in charge of the semantic analysis of amendment provisions as well as by completing the functions to support the consolidation process. Note that some manual efforts will be always needed to verify the results of the automatic facilities and to cope with some inaccuracies of the legislator himself.

The following extensions of this work will involve:

- the recognition of the application date wording and the extraction of the related metadata;
- the analysis of multiple text modifications, namely amendments concerning more than one partition or word occurrence as well as combined amendments within the same provision text. In spite of the fact that such an extension can be complex, we believe that for particularly significant sub–types of provisions it can be successfully carried out. This is the case in which the same type of amendment is applied to different portions of the act (ex. “art. 5 and 7 of the act n. 1/2000 are repealed”), or in which different types of amendement apply to the same act (ex. “in the act 1/2000, art. 5 is repealed and art. 7 is substituted by the following ...”);
- the integration, with possible adaptations, in the Editor of other tools developed in the Italian standards environment: an intelligent XML differences extractor JNDiff (to detect and show only the differences between two versions of the same act) and a parallel text formatter TafWeb (to highlight the partitions which are unchanged or have been changed between two versions)[3].

[3]JNDiff and TaFWeb are open source software developed for the Italian Senate to produce the differences and the parallel text between the versions of the same bill approved by the two Parliament branches.

Results obtained so far are encouraging in terms of the quality and robustness of the current implementation. However, there is clearly more work needed for this metadata extraction prototype to be extensively used on large law text corpora.

8. REFERENCES

[1] M. Seppius. Consolidation, *Interim report of the Working Group of the European Forum of Official Gazettes* available at circa.europa.eu/irc/opoce/ojf/info/data/prod/html/Consolidation%20-%20interim%20report.doc, September 2008.

[2] Y. Ogawa, S. Inagaki, and K. Toyama. Automatic Consolidation of Japanese Statutes based on Formalization of Amendment Sentences, In K. Satoh et al. (eds.), *JSAI 2007*, LNAI 4914, Springer-Verlag Berlin Heidelberg, pp. 363-376.

[3] R. Bartolini, A. Lenci, S. Montemagni, V. Pirrelli, and C. Soria. Automatic classification and analysis of provisions in italian legal texts: a case study. In *Proceedings of the Second International Workshop on Regulatory Ontologies*, 2004.

[4] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria. Automatic semantics extraction in law documents. In *Proceedings of International Conference on Artificial Intelligence and Law*, June 6-11 2005, Bologna, pp. 133-140.

[5] T. Agnoloni, E. Francesconi, and P. Spinosa. , xmLegesEditor: an OpenSource Visual XML Editor for supporting Legal National Standards. In *Proceedings of the V Legislative XML Workshop*(2007), pp. 239-251.

[6] S. Federici, S. Montemagni, and V. Pirrelli. Shallow Parsing and Text Chunking: a View on Underspecification in Syntax. In *Proceedings of the Workshop On Robust Parsing*, in the framework of the European Summer School on Language, Logic and Information (ESSLLI-96), Prague, 1996.

[7] R. Brighi, L. Lesmo, A. Mazzei, M. Palmirani, and D. P.Radicioni. Towards Semantic Interpretation of Legal Modifications through Deep Syntactic Analysis. In *Proceedings of the 21th International Conference on Legal Knowledge and Information Systems (JURIX 2008)*, 10-13 December, Florence, 2008.

[8] X. Li, and D. Roth. Exploring Evidence for Shallow Parsing. In *Proceedings of the Annual Conference on Computational Natural Language Learning* , Toulouse, France, 2001.

[9] C. Grover, B. Hachely, I. Hughson, and C. Korycinski. Automatic Summarisation of Legal Documents. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law (ICAIL 2003)*, Scotland, United Kingdom, 2003.

[10] R. Bartolini, A. Lenci, S. Montemagni, and V. Pirrelli. Hybrid Constrains for Robust Parsing: First Experiments and Evaluation. In *Proceedings of International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, 2004.

[11] G. Venturi. Parsing Legal Texts. A Contrastive Study with a View to Knowledge Management Applications. In *Proceedings of Sixth International Conference on Language Resources and Evaluation (LREC 2008)*,

Workshop *Semantic Processing of Legal Texts*, Marrakech, Morocco, May 26-1 June 2008, CD-ROM.

[12] F. Erik, Tjong Kim Sang, and S. Buchholz. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000.

[13] J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

[14] T. Arnold-Moore. Automatic generation of amendment legislation. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law (ICAIL 1997)*, pp. 56-62.