

Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

**IL MARKUP AUTOMATICO SU DOCUMENTI
LEGISLATIVI IN LINGUA ALBANESE**

Relatore: Chiar.mo

Prof. FABIO VITALI

Presentata da:

NICOLA BINA

Sessione I Anno Accademico 2022/2023

Sommario

La presente dissertazione si basa sull'aggiunta di una funzionalità all'interno dell'editor LIME, la cui principale funzione è il trattamento di documenti legislativi redatti nello standard Akoma Ntoso, basato sul linguaggio XML. Tale funzionalità implementata riguarda l'estensione della funzione di markup automatico anche ai documenti legislativi Albanesi, essa era già presente per i testi in lingua italiana, inglese, spagnola, russa e rumena. Tale funzione utilizza parser basati su espressioni regex permettono la rilevazione automatica degli elementi presenti all'interno del testo legislativo. All'interno della dissertazione verrà spiegato nel dettaglio il funzionamento di tale funzione di markup automatico, dei relativi parser e delle relative espressioni regolari utilizzate da essi per rilevazione degli elementi all'interno dei documenti.

Indice

Sommario.....	1
Indice.....	2
Elenco figure.....	4
1 Introduzione.....	6
2 La rappresentazione digitale dei documenti legislativi.....	11
2.1 I documenti legislativi nell'era digitale.....	11
2.2 La struttura e il markup dei documenti.....	12
2.3 Akoma Ntoso.....	14
2.4 LIME.....	15
2.4.1 Presentazione.....	15
2.4.2 Creazione ed importazione del documento.....	17
2.4.3 Markup automatico.....	18
2.4.4 Markup manuale.....	19
2.4.5 Validazione del codice XML.....	22
2.4.6 Salvataggio ed esportazione del documento.....	23
2.5 La lingua albanese e la sua aggiunta in LIME.....	24
3 Gli elementi di markup in lingua albanese.....	25
3.1 Gli elementi di markup.....	26
3.1.1 Intestazione.....	27
3.1.2 Preambolo.....	27
3.1.3 Corpo.....	28
3.1.4 Conclusione di un documento.....	31
4 Configurazione di LIME per la lingua albanese.....	33
4.1 Customizzazione della lingua nell'ambiente.....	33
4.2 Una nuova lingua nel markup automatico.....	35

4.2.1 Struttura e analisi di un file conf.....	38
5 Valutazioni.....	41
5.1 Uno sguardo d'insieme.....	41
5.2 Il problema dei riferimenti.....	43
5.3 Il problema delle citazioni.....	44
6 Conclusioni.....	46
Bibliografia.....	49

Elenco figure

1.1 Editor LIME con documento non marcato.....	7
1.2 Editor LIME con documento marcato.....	9
2.1 Logo di Akoma Ntoso.....	14
2.2 Editor LIME.....	16
2.3 Menu dove inserire le caratteristiche di un documento.....	17
2.4 Menu di importazione.....	18
2.5 Pannello degli strumenti.....	20
2.6 Menu di cancellazione.....	20
2.7 Outline.....	21
2.8 Messaggio di errore nel documento XML.....	22
2.9 Messaggio di validazione del documento XML.....	23
2.10 Menù di salvataggio del documento.....	23
3.1 Documento in lingua spagnola.....	25
3.2 Documento in lingua albanese.....	26
3.3 Intestazione di un documento.....	27
3.3 Preambolo di un documento.....	27
3.4 Parte del corpo di un documento.....	28
3.5 Riferimento con pannello per l'inserimento dei dati.....	30
3.6 Un articolo contenente due quoted text.....	30
3.7 Una quoted Structure nel quale viene citato un elemento esterno.....	31
3.8 Conclusioni di un documento.....	31
4.1 Menu delle preferenze per la scelta della lingua.....	33
4.2 Il file contenente le lingue selezionabili.....	34
4.3 Un file Strings.js per la customizzazione della lingua.....	35
4.4 Cartelle contenenti i parser e le espressioni regolari per il markup automatico.....	36
4.5 Interno di una cartella avente le funzioni di parsing.....	37
4.6 Contenuto del file conf.php del parser structure.....	38

5.1 Documento la cui struttura e gli elementi vengono catturati dalla sola funzione di markup automatico.....	41
5.2 Riferimento non elaborato correttamente.....	43
5.3 Citazioni annidate marcate in maniera scorretta.....	45

Capitolo 1

Introduzione

All'interno delle società, le leggi sono sempre state il fondamento che ne regolamentava il modo di vivere e comportarsi delle persone, delle istituzioni e delle organizzazioni, al fine di assicurare un giusto equilibrio tra i diritti individuali e le esigenze della società nel suo complesso.

Fino a poco tempo fa, le informazioni di carattere giuridico venivano principalmente veicolate attraverso documenti scritti su carta, i quali erano creati da esperti del dominio legislativo. Tale categoria di professionisti era l'unica che disponeva delle risorse materiali necessarie per trattare e consultare tali testi.

Grazie all'arrivo di nuove tecnologie e alla loro diffusione tramite Internet, è ora possibile digitalizzare varie risorse che possono essere facilmente gestite, trattate e consultate a distanza. Ciò offre opportunità di accesso alle informazioni a un numero sempre maggiore di persone anche non appartenenti alla categoria di esperti, che possono accedervi da qualsiasi luogo e in qualsiasi momento. Gli addetti ai lavori hanno a disposizione una sempre maggiore varietà di strumenti per la trattazione e la consultazione efficace per la loro professione di tali testi legislativi.

Lo scopo di questa dissertazione è raccogliere automaticamente i dati sulla struttura ed il contenuto di documenti giuridici in una lingua non ancora implementata all'interno di un editor online, e infine mostrare gli elementi trovati all'interno del testo all'interno editor il quale rende disponibili ulteriori strumenti per la trattazione dei documenti inseriti.

Utilizzando l'editor di markup LIME è possibile raccogliere tali dati riguardanti la struttura dei documenti legislativi nella lingua nel quale il documento legislativo viene redatto e di mostrarli con l'utilizzo del linguaggio Akoma nToso (basato sul linguaggio XML) in maniera automatica via web tramite l'impiego di un browser.

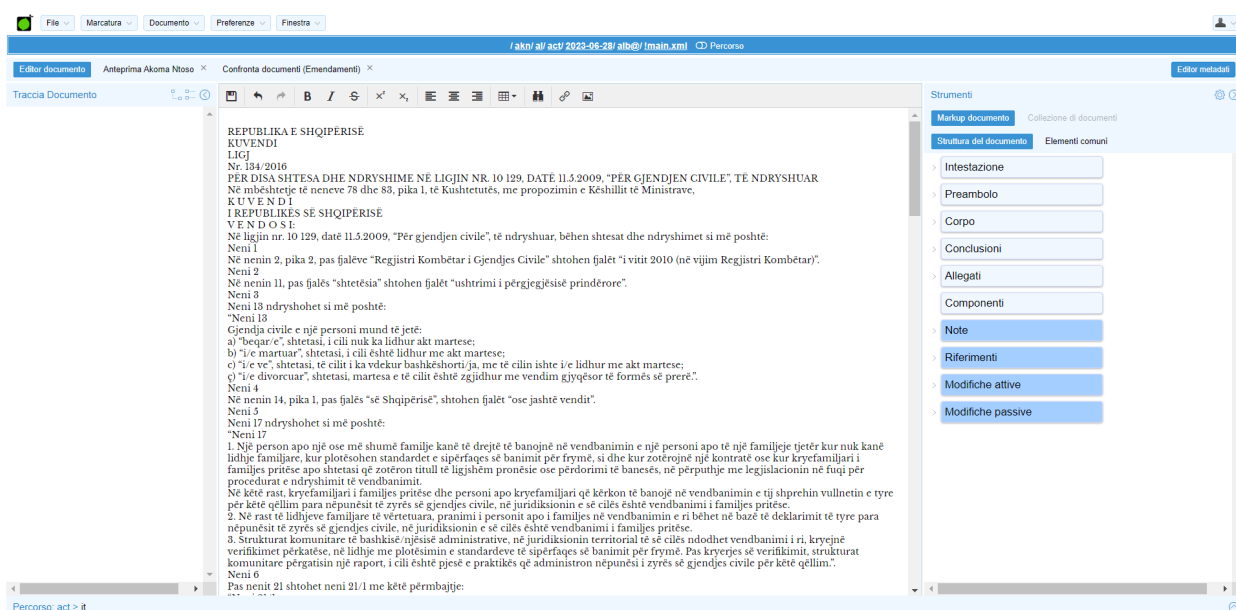


Figura 1.1 Editor LIME con documento non marcato

L'editor LIME permette, scegliendo tipologia del documento e lingua, di analizzare tramite un parser un documento. Il parser riconosce al suo interno gli elementi costitutivi, così da poterli classificarli in sezioni diverse.

In precedenza, LIME veniva utilizzato per l'analisi dei documenti in lingua italiana, inglese, spagnola, russa, e rumena. Su richiesta del Parlamento albanese, si è estesa tale funzionalità anche ai documenti in lingua albanese.

Sono state aggiunte le funzionalità per poter processare i documenti legislativi in lingua albanese e per riconoscerne e marcare al suo interno i macroelementi che identificano la struttura del documento:

- prefazione;
- preambolo;
- corpo;
- conclusioni.

Così come gli elementi ad essi interni, nella prefazione:

- tipo del documento;
- numero del documento.

Nel preambolo:

- formula di emanazione.

Nel corpo:

- titoli;
- sezioni;
- articoli;
- paragrafi;
- liste;
- punti;
- testo citato;
- riferimenti.

Nelle conclusioni:

- data del documento;
- firma.

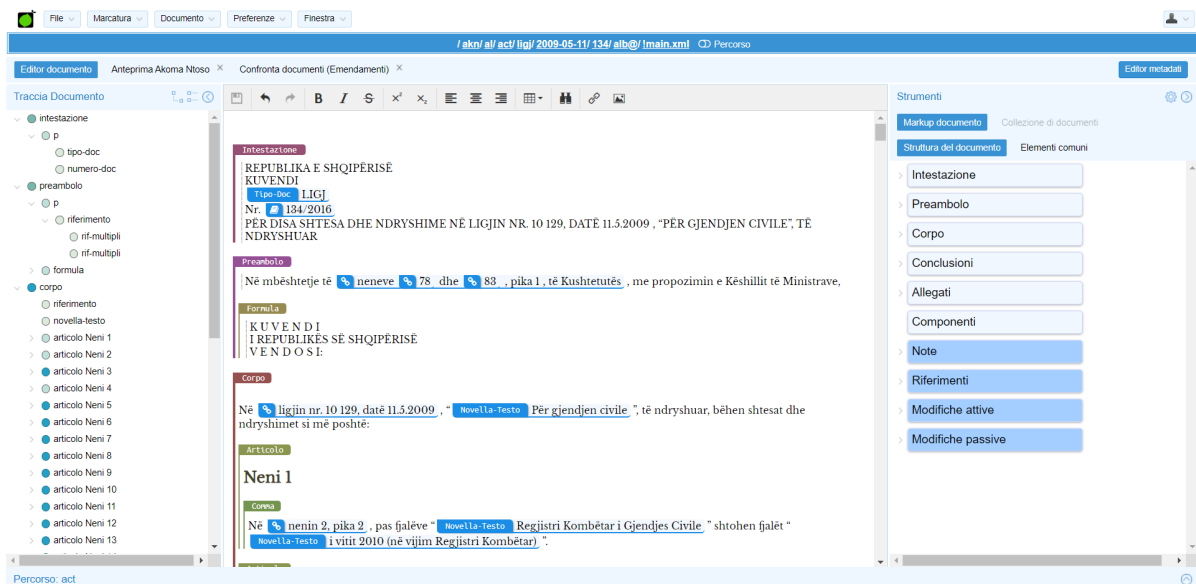


Figura 1.2 Editor LIME con documento marcato

Questa dissertazione è così suddivisa:

- Il capitolo 2 mostra una breve storia della digitalizzazione dei documenti legislativi e dei linguaggi di markup, una descrizione del linguaggio XML e del formato Akoma Ntoso e dell'editor LIME;
- Il capitolo 3 fornisce una panoramica sugli elementi di markup che vengono evidenziati all'interno di documento legislativo e delle caratteristiche che presentano all'interno di tale tipologia di documenti in lingua albanese;
- Il capitolo 4 si occupa di mostrare i dettagli tecnici dell'implementazione, in modo da agevolare la comprensione del codice e consentirne la modifica ai futuri sviluppatori. In particolare, la documentazione si concentra su come aggiungere una nuova lingua per operare all'interno dell'ambiente LIME e di modificare le espressioni regolari utilizzate dai parser per la rilevazione automatica degli elementi interni ai documenti.

- Il capitolo 5 mostra un'analisi critica sul lavoro svolto per l'aggiunta della funzione di markup automatico in lingua albanese, sulle mancanze presenti e sulle possibili soluzioni che possono essere applicate.
- Il capitolo 6 contiene le conclusioni e i possibili sviluppi futuri per lo strumento realizzato.

Capitolo 2

La rappresentazione digitale dei documenti legislativi

2.1 I documenti legislativi nell'era digitale

I documenti giuridici sono testi, come ad esempio una legge, un emendamento o una sentenza, promulgati da un ente legale autorevole in un determinato paese, come un Parlamento, una Camera o un Tribunale. Con la nascita dei calcolatori elettronici negli anni '50 e '60, si è creata la branca di studio dell'Informatica Giuridica i cui scopi riguardano anche il trattamento di tali documenti giuridici in forma digitale all'interno dei neonati database. A partire dal 1970 vi fu notevole aumento dei documenti generati dai vari organi giuridici; tale tendenza continuò per tutto il periodo degli anni '80, durante tale periodo iniziarono anche a diffondersi i personal computer, i quali a partire dalla fine degli anni '90 poterono iniziare a connettersi tra loro e a scambiarsi informazioni tramite Internet. Tale avvento di Internet ha radicalmente cambiato i modi in cui l'informazione è prodotta e comunicata e ha permesso negli anni 2000 a sempre più persone di ottenere l'accesso a documenti legali di ogni genere. Pertanto, l'importanza dei sistemi informativi legislativi è notevolmente aumentata in quanto sono diventati fondamentali sia per la comunicazione tra i vari apparati legali e i professionisti del settore, sia per l'e-government, ossia il sistema di gestione digitale dell'amministrazione pubblica, il quale, erogando informazioni e servizi, ha reso più trasparenti i processi legislativi e ha agevolato la partecipazione dei normali cittadini alla vita pubblica, rendendo il governo più responsabile, trasparente ed efficace.[BF08].

2.2 La struttura e il markup dei documenti

I documenti strutturati sono testi in formato elettronico il cui contenuto è composto dal semplice testo e da ulteriori elementi utilizzati per identificare la sua struttura o informazioni che si riferiscono a risorse interne o esterne al documento. I linguaggi di markup hanno permesso di definire e formalizzare tali elementi all'interno del documento.

Il linguaggio SGML (Standard Generalised Markup Language) è stato uno dei primi ad essere utilizzati a tale scopo. Da esso è poi stato definito nel 1998 il linguaggio XML (eXtensible Markup Language) [BPS97][BB+99].

L'HTML (HyperText Markup Language, derivato nel 1993 da SGML) è anch'esso un linguaggio di markup che permette nello specifico la creazione e lo sviluppo di siti e pagine navigabili all'interno del web.

Tuttavia, a differenza dell'HTML che è stato pensato con lo scopo di creare pagine web, l'XML permette la definizione di nuovi elementi e quindi di creare così sotto-linguaggi caratteristici utilizzati per scopi particolari [LS99] come ad esempio il TEI markup language per la trattazione dei testi letterari oppure MathML per la descrizione di notazioni matematiche.

Un esempio del linguaggio XML utilizzato per definire la disposizione di un documento strutturato con vari elementi al suo interno:

```
<documento>
  <titoloDOC> Titolo Documento XML </titoloDOC>
  <indice>lista dei contenuti</indice>
  <capitolo>
    <numero>1</numero>
    <titolo> Introduzione del documento </titolo>
    <contenuto> Contenuto del capitolo </contenuto>
  </capitolo>
  ...
  <bibliografia> Lista delle fonti </bibliografia>
</documento>
```

Dunque, il linguaggio XML risulta essere una soluzione adeguata alla descrizione di documenti strutturati riguardanti l'ambito giuridico. Tuttavia, i documenti legislativi possono variare molto se si cambia nazione, epoca della pubblicazione, l'istituzione che lo pubblica, e la tipologia del documento. Si deve inoltre considerare che al loro interno vi possono essere riferimenti ad altri documenti a loro volta interconnessi tra di loro, e il fatto che vi possono essere più versioni da gestire dello stesso documento.

Vi sono un grosso numero di standard XML creati per le varie tipologie di documenti giuridici[PV12]. Tuttavia, spesso sono fatti per marcare una singola tipologia o le leggi di un singolo stato. Lo standard Akoma Ntoso è agnostico rispetto alla nazione, tipologia, epoca o istituzione e permette di essere utilizzato per la creazione di qualunque documento legato al mondo giuridico.

2.3 Akoma Ntoso

Lo standard XML Akoma Ntoso (Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies) è stato sviluppato nel 2005 nell'ambito di un progetto del Dipartimento delle Nazioni Unite per l'Economia e gli Affari Sociali il cui scopo era quello di aiutare le legislature africane a realizzare meglio le loro funzioni democratiche utilizzando le tecnologie dell'informazione [VZ07][BCP09].

In figura 2.1, il logo di Akoma Ntoso è un simbolo che sta a significare concordia e armonia, (Akoma Ntoso significa, in italiano, "cuori uniti") ed è utilizzato dal popolo Akan dell'Africa occidentale per indicare la comprensione e l'accordo[PV11].

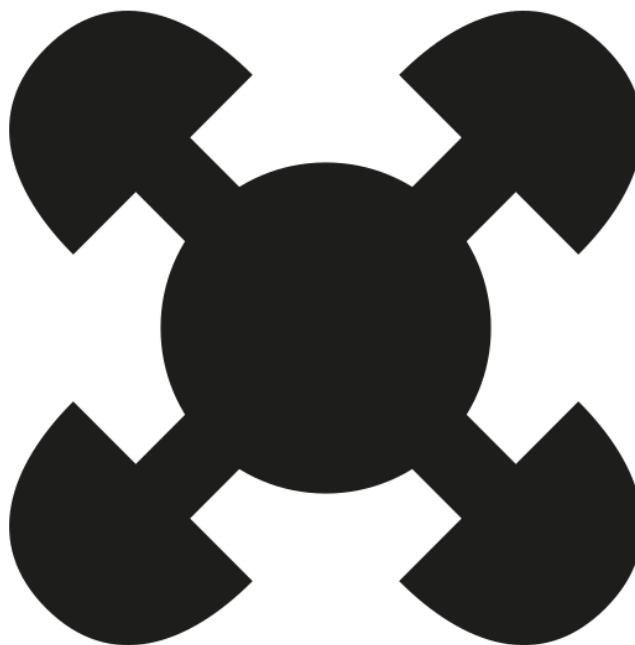


Figura 2.1 Logo di Akoma Ntoso

Esso è pensato per essere universale all'interno dell'ambito giuridico e dunque non orientato ad un singolo contesto legislativo. Inoltre, i documenti così creati hanno il vantaggio di poter essere compresi ed analizzati sia dagli esseri umani (human-readable) che da un programma informatico (machine-readable) entrando dunque a fare parte del Web Semantico rispettandone i requisiti così come indicato dal suo ideatore Berners-Lee[B99].

Lo standard Akoma Ntoso è basato sui pattern (il risultato derivante dall'estrazione e dalla distillazione di esperienze precedenti), i quali permettono di definire un piccolo numero di classi di elementi che interagiscono tra loro all'interno del documento in una maniera molto specifica [BPV11].

2.4 LIME

LIME (Language Independent Markup Editor) è un web editor di markup open-source, sviluppato dal CIRSFID e dall'Università di Bologna inizialmente per il trattamento dei documenti prodotti dal Parlamento italiano. LIME utilizza Akoma Ntoso per descrivere documenti legali ed amministrativi in maniera strutturata, permette la registrazione degli utenti ed il salvataggio dei documenti in cloud, utilizzando un'interfaccia del tutto simile ai più utilizzati programmi di editing testuali, e.g. Microsoft Word [C13].

2.4.1 Presentazione

L'applicazione presenta le seguenti componenti:

- In alto, troviamo la barra degli strumenti per le preferenze dell'editor e per il trattamento e la visualizzazione del documento sul quale si vuole lavorare.

- A sinistra è presente l'outline nel quale viene mostrata la struttura e gli elementi rilevati.
- Al centro è presente l'editor nel quale viene visualizzato il documento e le marcature su di esso effettuate.
- A destra sono presenti gli strumenti da utilizzare per marcare il testo con gli elementi dello standard Akoma Ntoso.
- In basso è presente l'editor per la compilazione dei metadati.

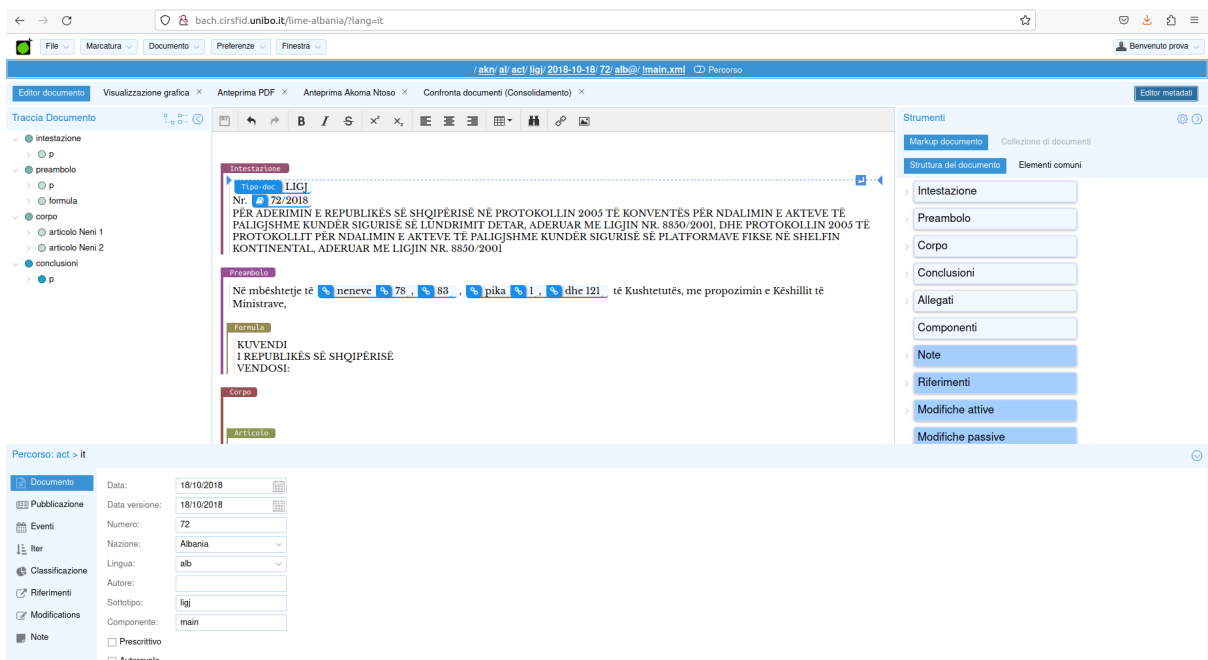
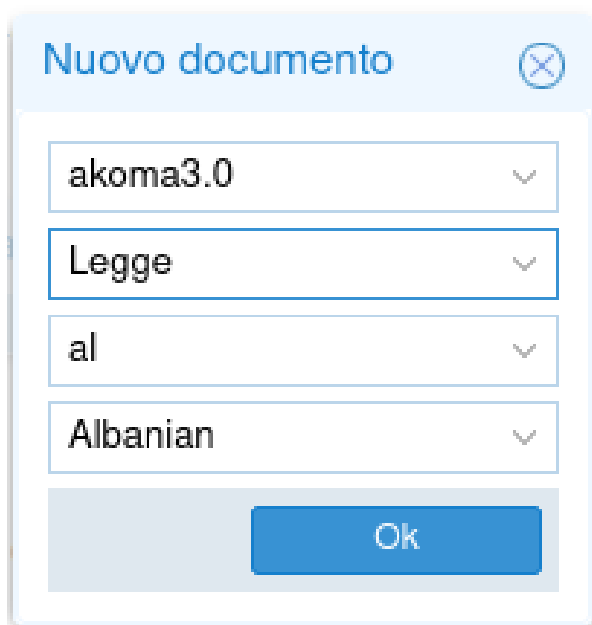


Figura 2.2 Editor LIME

2.4.2 Creazione ed importazione del documento

Per creare un documento all'interno dell'ambiente LIME, bisogna cliccare sul tasto "FILE", poi selezionare la voce "Nuovo documento". Dal menù vengono indicate la lingua nel quale è scritto il documento insieme al codice ISO della nazione, il linguaggio di markup da utilizzare e il tipo di documento da trattare (Figura 2.3). Si può dunque iniziare a scrivere il testo del nuovo documento.



The image shows a dialog box titled "Nuovo documento" with a close button in the top right corner. It contains four dropdown menus stacked vertically. The first dropdown menu is set to "akoma3.0", the second to "Legge", the third to "al", and the fourth to "Albanian". At the bottom right of the dialog is a blue button labeled "Ok".

Figura 2.3 Menu dove inserire le caratteristiche di un documento

Se invece si vuole importare un documento già pronto dall'esterno, sempre cliccando sul tasto "FILE", bisogna selezionare la voce "Importa documento". A quel punto, si aprirà il menù di importazione (Figura 2.4) nel quale si potrà scegliere il file contenente il documento da importare (esplorando le cartelle e i file del pc, o utilizzando la funzione di drag and drop).

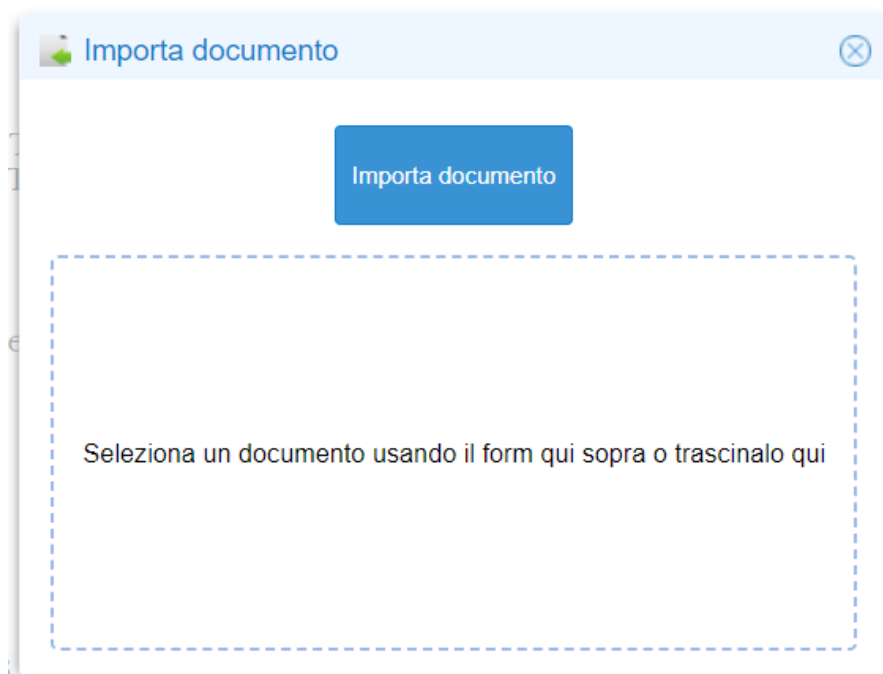


Figura 2.4 Menu di importazione

Dopodiché, bisogna compilare i campi del menù già descritto in precedenza nel quale si descrivono le caratteristiche del documento (Figura 2.3) ed avremo il documento scelto all'interno dell'editor pronto per essere marcato.

2.4.3 Markup automatico

Per iniziare a trattare un documento ed evidenziarne gli elementi al suo interno, è possibile utilizzare la funzione di markup automatico. Tale funzione permette di ricercare le parole o frasi indicanti elementi normativi presenti all'interno del testo nella lingua che viene rilevata o indicata nel momento della creazione o dell'importazione del documento.

Per la ricerca di tali elementi normativi avviene tramite l'utilizzo di parser, ossia un programma che scansiona il testo del documento e ne ricerca gli elementi da evidenziare in esso presenti. Tale programma di parsing utilizza per la rilevazione di ciascuno dei possibili elementi delle espressioni regolari.

Le espressioni regolari (RegEx) sono sequenze di caratteri speciali utilizzate per cercare e manipolare specifiche porzioni di testo all'interno di una stringa più grande, definiscono una funzione che, ricevendo come input una stringa, restituisce un valore booleano (sì/no) a seconda che la stringa segua o meno un determinato pattern. Esse forniscono uno strumento di ricerca e analisi del testo, trovando specifiche parole, numeri, date o pattern complessi [F12].

Tali regex vengono utilizzate fin dal 1968 sia per il pattern matching[T68] che per l'analisi lessicale [JPA68]. Il loro utilizzo si è diffuso negli anni '80 quando hanno iniziato a diffondersi i primi linguaggi di markup, come SGML.

Al giorno d'oggi, le espressioni regolari vengono ampiamente utilizzate nei linguaggi di programmazione, esse infatti fanno parte della libreria standard di molti linguaggi inclusi Java e Python, oltre che ad essere utilizzate nei programmi di elaborazione del testo e in diversi editor.

2.4.4 Markup manuale

È possibile utilizzare il pannello degli strumenti (posizionato a destra dell'editor di LIME) per marcare manualmente il testo con ulteriori nel caso vi siano elementi non rilevati dal parser per la marcatura automatica arricchendo con ulteriori dettagli il documento trattato.

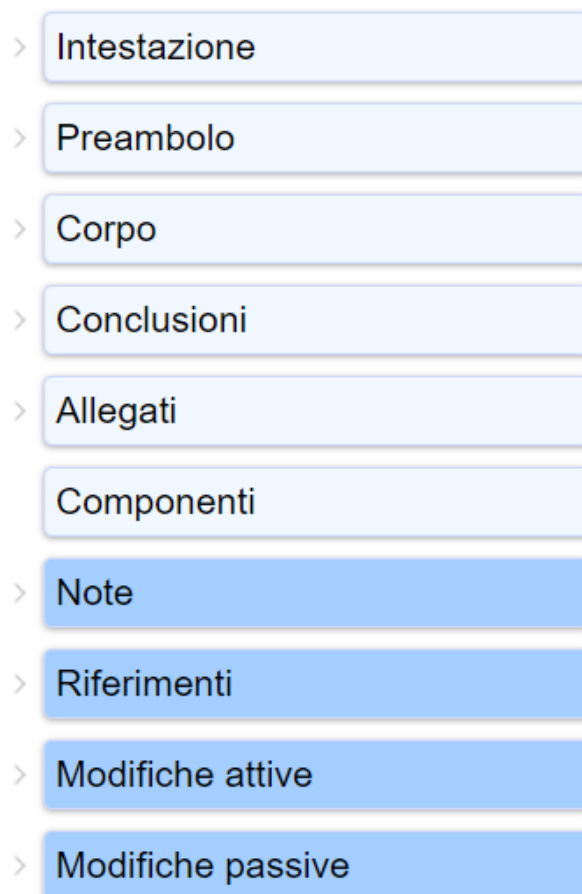


Figura 2.5 Pannello degli strumenti

Inoltre è possibile che durante il processo di marcatura (sia manuale che automatica) vengano rilevati degli elementi errati. Per eliminare un elemento indesiderato, basta cliccare con il tasto destro sul componente e utilizzare il menu di cancellazione.

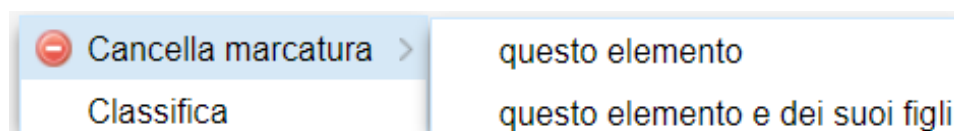


Figura 2.6 Menu di cancellazione

Tale menù permette di scegliere tra due opzioni di cancellazione:

- eliminazione del singolo elemento selezionato, lasciando inalterati gli altri eventuali componenti annidati al suo interno.
- eliminazione dell'elemento insieme a tutta la struttura composta dagli elementi contenuti al suo interno.

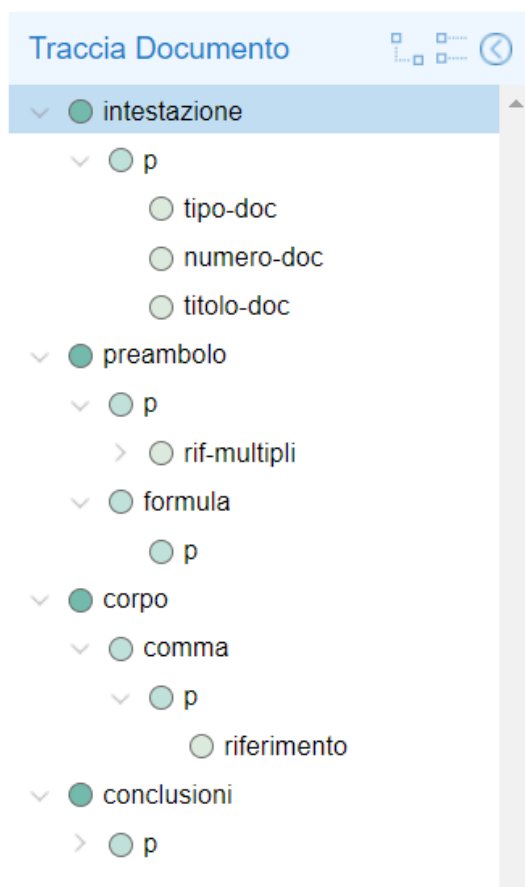


Figura 2.7 Outline

Aggiungendo ed eliminando gli elementi interni al documento né si altera la struttura. La quale può essere visualizzata ed esplorata tramite l'utilizzo dell'outline posta a sinistra dell'editor.

2.4.5 Validazione del codice XML

Una volta terminato il lavoro di marcatura all'interno del documento sul quale si sta operando si può controllare il codice XML prodotto utilizzando il bottone “anteprima Akoma Ntoso”. Dopodichè si deve validare il documento XML prodotto(per provarla cliccare il bottone “documento” e selezionare la voce “Validazione XML”). Tale validazione controlla che la struttura del documento XML non abbia errori al suo interno, e se presenti una volta rilevati essi vengono mostrati tramite relativi messaggi di errore(Figura 2.8), altrimenti verrà mostrato un messaggio indicante la correttezza interna del documento XML prodotto(Figura 2.9).

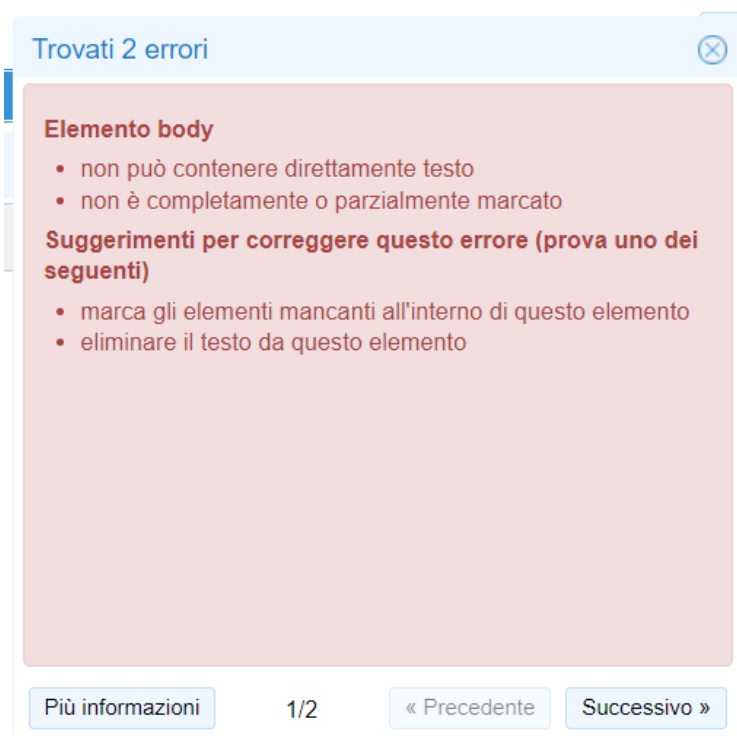


Figura 2.8 Messaggio di errore nel documento XML

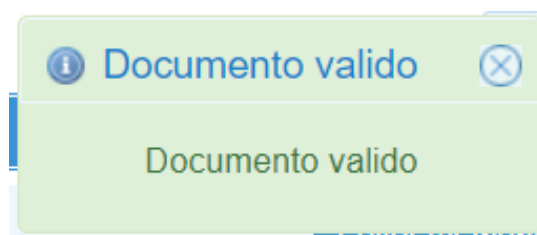


Figura 2.9 Messaggio di validazione del documento XML

2.4.6 Salvataggio ed esportazione del documento

Se si vuole interrompere il lavoro di marcatura sul documento è sufficiente salvare il documento sull'account LIME con il quali si è loggati in quel momento cliccando sul tasto “FILE” e selezionare la voce “salva come” e procedere con l’indicare le caratteristiche del documento che poi andranno a costituire il percorso di salvataggio del documento all’interno dell’account nel quale si sta lavorando, tali caratteristiche sono in ordine: nazione del documento, tipo del documento, sottotipo (corrisponde all’elemento di tipo Tipo Doc.), data (corrisponde all’elemento di tipo Data Doc.), numero (corrisponde all’elemento di tipo Numero Doc.), versione, e nome.

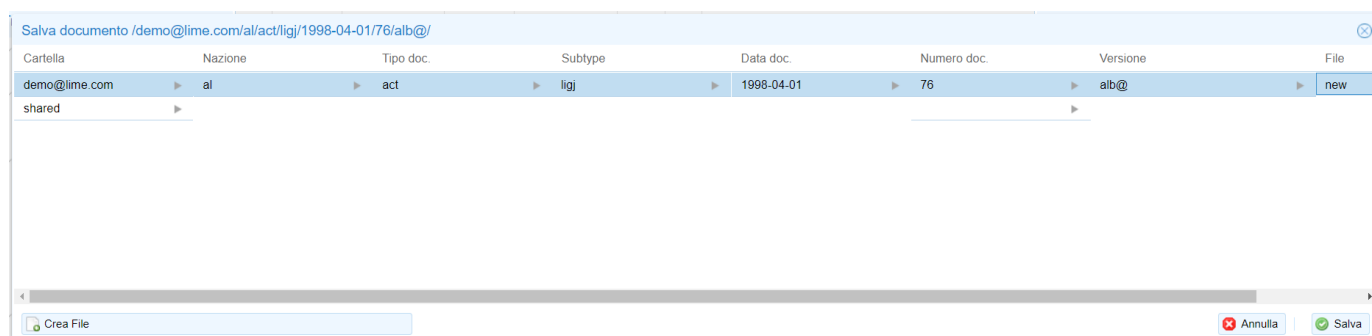


Figura 2.10 Menù di salvataggio del documento

Una volta salvato il file, è possibile recuperarlo premendo sul tasto “FILE” e cliccare la voce “Apri documento” e selezionare il file desiderato.

Per esportare invece in locale il documento presente nell'editor cliccare sul tasto "FILE" e cliccare la voce "Esporta come..." e scegliere il formato nel quale si vuole salvare il testo (si raccomanda di scegliere l'XML), per continuare a lavorare all'interno di LIME con il documento così salvato sul pc seguire le istruzioni per l'importazione di un documento.

2.5 La lingua albanese e la sua aggiunta in LIME

La lingua albanese (*gjuha shqipe*) è una lingua della famiglia indoeuropea. La lingua albanese unificata o "standard" è basata principalmente sulla variante tosco, con alcuni elementi dell'altra variante, il ghego. L'albanese utilizza l'alfabeto latino, al quale aggiunge le lettere *ë*, *ç*. Per altri fonemi non presenti nell'alfabeto latino, utilizza 9 digrammi aggiuntivi. È parlata da circa 7,6 milioni di persone principalmente in Albania, della quale è lingua ufficiale.

Dal 2021 esiste un progetto di collaborazione tra l'università di Bologna e il Parlamento Albanese per lo sviluppo di una personalizzazione dello standard di Akoma Ntoso che raggiunga i seguenti obiettivi:

- Una estensione che tratti i documenti giuridici albanesi;
- Una personalizzazione della naming convention per gestire gli IRI delle leggi albanesi;
- Estendere le funzioni dell'editor LIME sia a livello di interfaccia che di servizi per supportare la lingua albanese.

Nel seguente testo si è potuta realizzare una soluzione al terzo obiettivo aggiungendo allo strumento di LIME una customizzazione dell'interfaccia e la funzione di markup automatico in lingua albanese.

Capitolo 3

Gli elementi di markup in lingua albanese

A seconda della tipologia, della nazione di provenienza e della lingua nel quale è scritto, un documento legislativo presenta uno stile interno differente rispetto agli altri documenti scritti per un differente scopo o in un'altra lingua.

The image shows a vertical document with several sections, each marked with a colored label on the left side. The labels are: 'Intestazione' (purple), 'Preambolo' (purple), 'Corpo' (red), 'Articolo' (green), and 'Comma' (green). The text is in Spanish and includes the following content:

- Intestazione:** Y PROGRAMA DE INGRESO CIUDADANO
- Preambolo:** Creación
TEXTO APROBADO
- Corpo:** (This label is positioned at the top of a large vertical line that runs down the page.)
- Articolo:** Artículo 1°.
- Comma:** (Emergencia Social).- Declárase la situación de emergencia social como consecuencia de las dificultades de inserción social comprobada por los indicadores de pobreza e indigencia constatados por el Instituto Nacional de Estadística.
- Articolo:** Artículo 2°.
- Comma:** (Plan de Atención).- Para la atención de la emergencia referida en el artículo anterior se establece el Plan de Atención Nacional de la Emergencia Social cuya vigencia se estima en dos años a partir de la promulgación

Figura 3.1 Documento in lingua spagnola

In figura 3.1 un documento in lingua spagnola che differisce dai documenti legislativi scritti in lingua albanese per la mancanza di una formula di emanazione all'interno del preambolo e di un diverso stile nell'indicare gli articoli di una legge.

3.1 Gli elementi di markup

Un documento legislativo è di solito suddiviso in varie sezioni; all'interno di ogni sezione, ci possono essere ulteriori sottosezioni o elementi, qui verranno analizzate ciascuna di tali componenti in particolare la forma che assumono all'interno di un documento legislativo albanese.

The image shows a screenshot of a legislative document in Albanian, with various sections and elements highlighted with colored boxes and labels. The document is titled "REPUBLICA E SHQIPËRIË KUVENDI" and is a law (LIGJ) numbered 69/2018. The document is divided into several sections: "Intestazione", "Preambolo", "Formula", "Corpo", "Articolo", and "Neni".

Intestazione
REPUBLICA E SHQIPËRIË
KUVENDI
Tipo-Doc: **LIGJ**
Nr. **69/2018**
PËRDISA SHITESA DHE NDRYSHIME NË LIGJIN NR. 10 129, DATË 11.5.2009 , "PËR GJENDJEN CIVILE", TË NDRYSHUAR

Preambolo
Në mbështetje të **neneve 78 , 81 , pika 1 , dhe 83 , pikat 1 e 2 , të Kushtetutës** , me propozimin e Këshillit të Ministrave dhe të një grupi deputetësh,

Formula
K U V E N D I
I R E P U B L I K Ë S S E S H Q I P Ë R I S E
V E N D O S I:

Corpo
Në **ligjin nr. 10129, datë 11.5.2009** , "**Novella-Testo Për gjendjen civile**" , të ndryshuar, bëhen këto shtesa dhe ndryshime:

Articolo
Neni 1
Comma
Në **nenin 2, pika 7** , pas fjalëve "**Novella-Testo janë emrat**" shtohen fjalët "**Novella-Testo dhe mbiemrat**" .

Articolo
Neni 2
Comma
Në **nenin 27, pika 3** ndryshohet si më poshtë:

Figura 3.2 Documento in lingua albanese

3.1.1 Intestazione

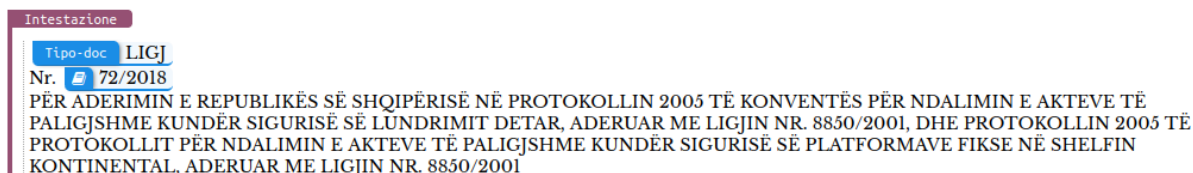


Figura 3.3 Intestazione di un documento

È la parte iniziale della legge al suo interno vengono rilevati questi due elementi:

-Tipo Doc.: indica la tipologia del documento legislativo, viene riconosciuto a partire da una lista nella quale vengono elencati i termini in albanese che identificano tale tipologia di documento.

-Numero Doc.: è il numero identificativo della legge emanata, tipicamente nelle leggi albanesi viene indicato nel seguente modo, numero-della-legge/anno-di-emanazione.

3.1.2 Preambolo

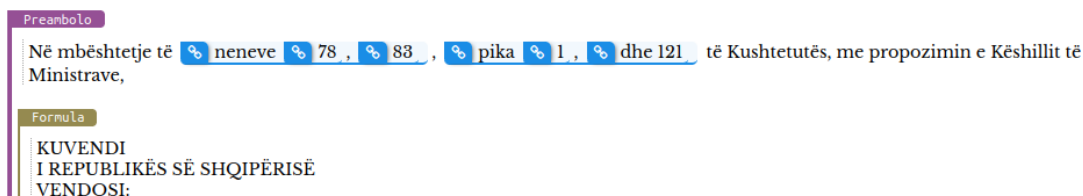


Figura 3.3 Preambolo di un documento

Contiene la premessa della legge, al suo interno è presente il seguente elemento ricorrente:

-Formula di emanazione: è la formula utilizzata per promulgare una nuova legge, viene rilevata dalle stesse regular expression utilizzate per dividere il preambolo dal corpo all'interno del documento.

3.1.3 Corpo

The image shows a vertical sidebar on the left with labels: "Capitolo", "Articolo", and "Contra" (repeated). The main content area displays the following structure:

- KREU III**
 - KUSHTET PËR RIPËRDORIM*
- Neni 9**
 - Formati i dokumenteve*
- 1.**

Organet e sektorit publik dhe sipërmarrjet publike vënë në dispozicion një dokument në formatin e gjuhën që zotërohet, në datën e kërkesës për ripërdorim e, kur është e mundur dhe e përshtatshme, përmes mjeteve elektronike në format të hapur, të lexueshëm nga kompjuterët, të aksesueshëm dhe të ripërdorshëm, së bashku me metadatat e tij.
- 2.**

Organet e sektorit publik dhe sipërmarrjet publike, kur është e mundur, hartojnë dhe bëjnë të disponueshme dokumentet për ripërdorim që nga faza e hartimit ose krijimit të tyre.
- 3.**

Organet e sektorit publik dhe sipërmarrjet publike nuk janë të detyruara të vijojnë prodhimin dhe ruajtjen e një lloji të caktuar dokumenti vetëm për ripërdorimin e tij nga një institucion i sektorit publik apo privat.
- 4.**

Organet e sektorit publik dhe sipërmarrjet publike, për aq sa është e mundur teknikisht, i bëjnë të disponueshme të dhënat për ripërdorim menjëherë pas mbledhjes, përmes ndërfaqeve të përshtatshme API, në formë dinamike ose kur është e përshtatshme, si një shkarkim i vetëm voluminoz(bulk download).

Figura 3.4 Parte del corpo di un documento

Nel corpo è presente il contenuto principale del documento, possono essere rilevati i seguenti elementi:

- Capitolo: Il testo di una legge può essere diviso in capitoli, seguiti da un valore numerico rappresentato come numero romano.
- Articolo: La legge è divisa in vari articoli ciascuno dei quali viene identificato da un valore numerico, può essere a sua volta suddiviso in vari paragrafi.
- Paragrafo: Un articolo può essere diviso al suo interno in più paragrafi numerati.
- Lista: Un elenco di punti opzionalmente preceduti da un'introduzione.
- Punti: I singoli elementi che insieme formano una lista. Possono essere di due tipi:
 - i punti delle liste normali indicati con le lettere dell'alfabeto albanese¹ seguiti dal carattere ")".
 - i punti delle liste contenute all'interno delle altre liste indicati con i numeri romani seguiti dal carattere ")".
- Riferimenti: riferimenti ad altre parti interne o esterne al testo del documento, ulteriori informazioni come la data e la nazionalità del documento possono venire visualizzati e modificati all'interno del menù sottostante che compare una volta cliccato sul riferimento, il markup automatico si occupa anche di compilare tale menù il quale dovrebbe venire inizializzato durante la fase di markup automatico, i riferimenti possono essere singoli, multipli o comprenderne un range indicato, possono inoltre stare ad indicare porzioni di testo interne od esterne al documento corrente e contenere ulteriori caratteristiche da rilevare.

¹ le lettere dell'alfabeto albanese sono le seguenti: a, b, c, ç, d, dh, e, ë, f, g, gj, h, i, j, k, l, ll, m, n, nj, o, p, q, r, rr, s, sh, t, th, u, v, x, xh, y, z, zh.

te [nenit 30, të ligjit nr. 9936, datë 26.6.2008](#)

Percorso: act > preamble > p > ref

Dati relativi al riferimento

Esterno	dynamic
Albania	Legge
ligj	Autore
26/06/2008	9936
Linguaggio	Data versione
art_30	

Salva

Figura 3.5 Riferimento con pannello per l'inserimento dei dati

- Citazioni: Porzioni di documenti esterne racchiuse tra una coppia di doppi apici. Possono essere di due tipi:
 - frase (*quoted text*): Una singola frase racchiusa all'interno di una coppia di doppi apici .

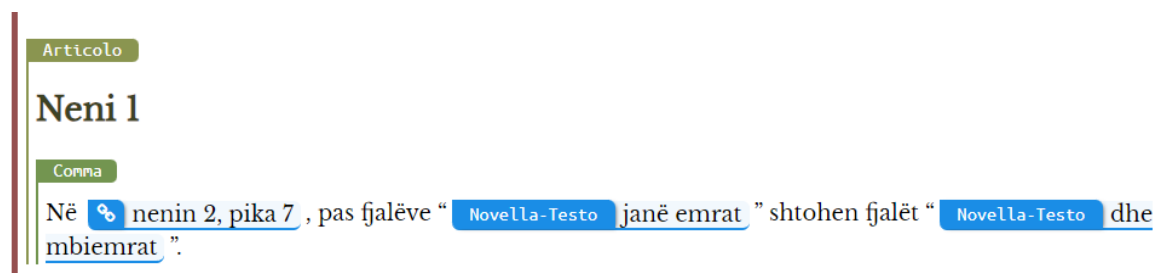


Figura 3.6 Un articolo contenente due *quoted text*

- citazione strutturata(*quoted structure*): La citazione interna ad coppia di doppi apici è composta da più frasi nelle quali è presente il carattere newline.

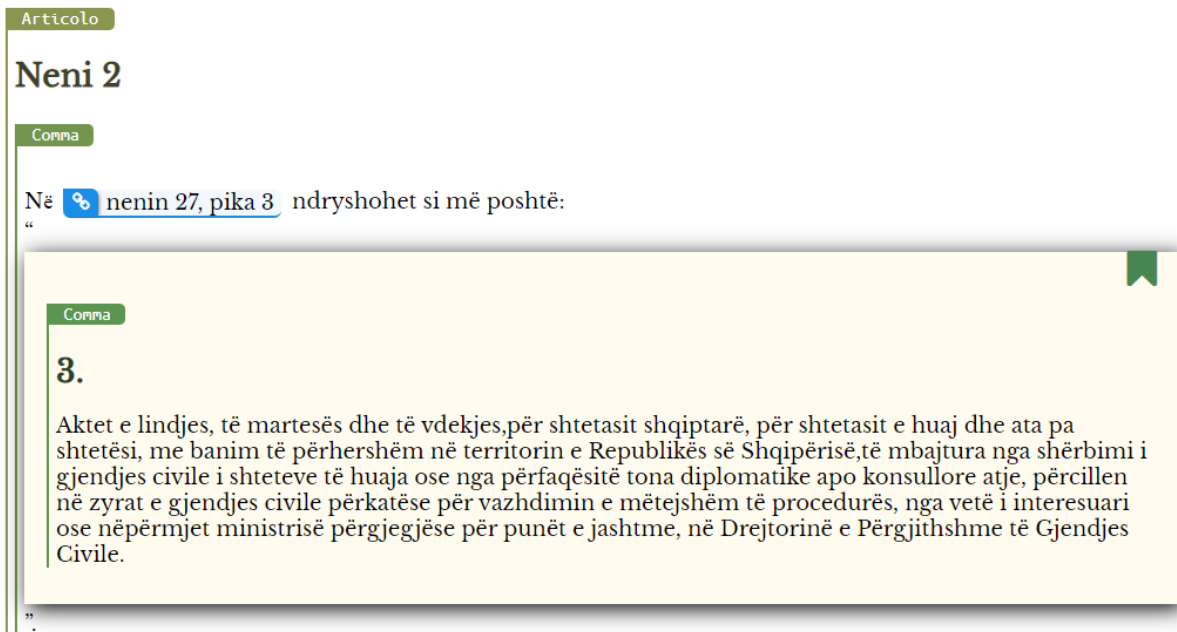


Figura 3.7 Una *quoted structure* nel quale viene citato un elemento esterno

3.1.4 Conclusione di un documento

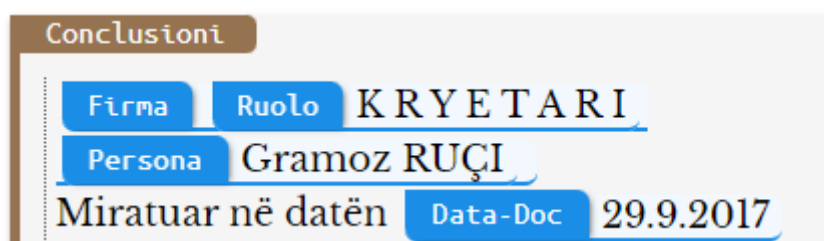


Figura 3.8 Conclusioni di un documento

Parte finale della legge, contiene questi due elementi:

- Data Doc.: indica la data di promulgazione della legge espressa in formato giorno/mese/anno.
- Firma: indica il nome di chi firma la legge appena approvata; opzionalmente può essere indicato anche il ruolo ricoperto dal firmatario.

Capitolo 4

Configurazione di LIME per la lingua albanese

Nel caso in cui sia necessario il trattamento di documenti scritti in una lingua non ancora presente nell'attuale versione di LIME, occorre aggiungere le espressioni regolari necessarie per il markup automatico dei documenti e le etichette che verranno inserite all'interno degli strumenti per la modifica e la trattazione del documento.

4.1 Customizzazione della lingua nell'ambiente

LIME permette all'utilizzatore di scegliere la lingua preferita dall'utente per operare all'interno dell'ambiente tramite il tasto delle preferenze.

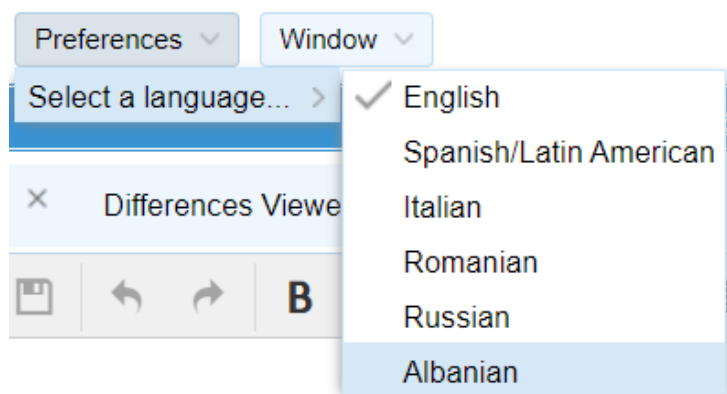


Figura 4.1 Menu delle preferenze per la scelta della lingua

Per aggiungere una lingua a questo elenco è necessario modificare l'array presente nel file:

```
“.../lime/packages/lime-core/src/store/Languages.js”
```

con l'oggetto contenente il nome del nuovo linguaggio da aggiungere e il corrispondente codice ISO a due lettere di tale linguaggio.

Può inoltre essere necessaria l'aggiunta del codice ISO indicante la nazione di provenienza del documento all'interno del file:

```
“.../languagesPlugins/akoma3.0/structure.json”
```

per le varie classi di documenti provenienti da tale nazione.

```
data : [
  { code : 'en', language : 'English' },
  { code : 'es', language : 'Spanish/Latin American' },
  { code : 'it', language : 'Italian' },
  { code : 'ro', language : 'Romanian' },
  { code : 'ru', language : 'Russian' },
  { code : 'al', language : 'Albanian' }
]
```

Figura 4.2 Il file contenente le lingue selezionabili

Infine, per permettere la traduzione effettiva degli elementi dell'editor di testo bisogna aggiungerne la traduzione nella nuova lingua all'interno dei vari file “Strings.js” presenti a partire dalla cartella packages ognuno dei quali si occupa della gestione di una determinata classe di nomi.

```

strings: {
  "en": {
    "importDocument": "Import document",
    "importDocumentTooltip" : "Import one of your files as a document",
    "selectDocument": "Select document",
    "selectDocumentExplanation": "Select a document using the form above or drag and drop your file here"
  },
  "it": {
    "importDocument": "Importa documento",
    "importDocumentTooltip" : "Importa un file come documento",
    "selectDocument": "Seleziona il documento",
    "selectDocumentExplanation": "Seleziona un documento usando il form qui sopra o trascinalo qui"
  },
  "es": {
    "importDocument": "Importar documento",
    "importDocumentTooltip" : "Puede importar un documento directamente desde sus archivos",
    "selectDocument": "Seleccione el documento",
    "selectDocumentExplanation": "Seleccione un documento utilizando el formulario de arriba o arrastrar y soltar el archivo aquí"
  },
  "ro": {
    "importDocument": "Importează un document",
    "importDocumentTooltip" : "Importează un fișier ca document",
    "selectDocument": "Selectează un document",
    "selectDocumentExplanation": "Selectează un document folosind formularul de mai sus sau Drag and Drop fisierul aici"
  },
  "ru": {
    "importDocument": "Импорт документа",
    "importDocumentTooltip" : "Импорт один из ваших файлов в качестве документа",
    "selectDocument": "Выберите документ",
    "selectDocumentExplanation": "Выберите документ, используя форму выше, или перетащить файл здесь."
  },
  "al": {
    "importDocument": "Importo dokumentin",
    "importDocumentTooltip" : "Importo nje dosje si dokument",
    "selectDocument": "Seleksiono dokumentin",
    "selectDocumentExplanation": "Seksiono nje dokument duke perdorur formen e mesiperme ose terhiqe ketu"
  }
}
}

```

Figura 4.3 Un file Strings.js per la customizzazione della lingua

4.2 Una nuova lingua nel markup automatico

Come già anticipato nel Capitolo 2, un parser è una funzione che scansiona il testo alla ricerca di elementi testuali presenti all'interno del documento. All'interno di LIME tali parser permettono di rilevare ciascuno in maniera automatica uno specifico elemento normativo all'interno del testo.

I parser di LIME vengono chiamati all'interno del file

“.../packages/akn-automatic-markup/src/Controller.js”

nel quale poi i risultati in seguito verranno poi trattati per marcare in maniera strutturata gli elementi rilevati. Tali parser sono allocati all'interno della cartella php/parsers.

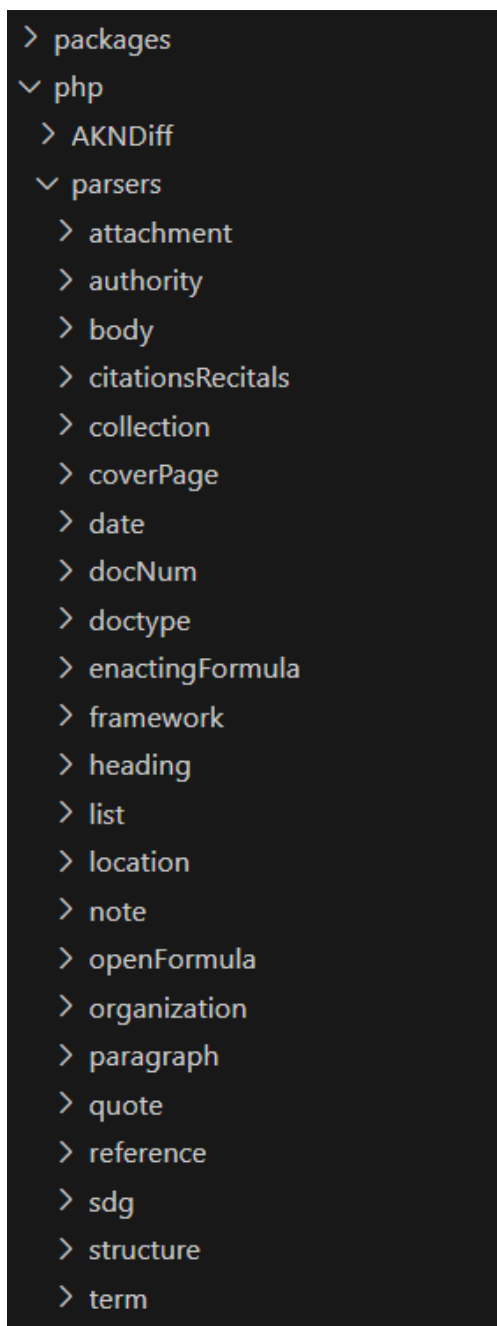


Figura 4.4 Cartelle contenenti i parser e le espressioni regolari per il markup automatico

Per ciascuno dei parser è presente una cartella nella quale tipicamente è presente il file che lo definisce ed una cartella ‘lang’ contenente le varie espressioni regolari da adottare a seconda del linguaggio in cui è stato scritto il documento da marcare.

Dunque, per implementare il markup automatico per una nuova lingua è necessario aggiungere per ciascuno dei parser che dovrà lavorare all’interno del documento una nuova cartella il cui nome è corrispondente al codice ISO 639-2 del linguaggio da implementare (nel caso dell’Albanese la cartella viene denominata “alb” è il codice ISO dell’Albanese), all’interno di tale cartella inoltre potrà essere necessaria la creazione di ulteriori cartelle indicanti il tipo di documenti che si vuole analizzare(nel nostro caso ci siamo limitati alla leggi dello stato albanese qui denominate come “act”).

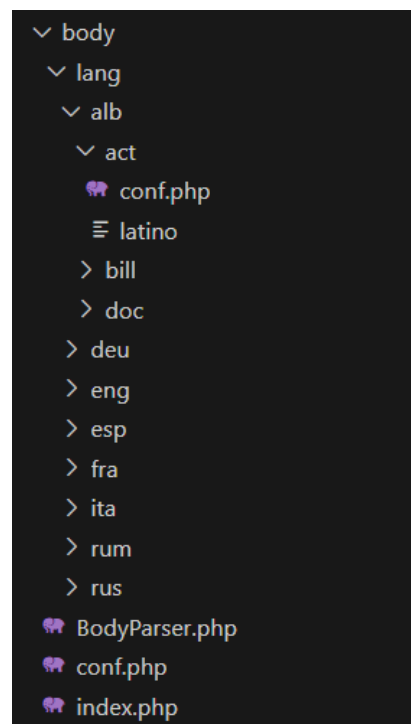


Figura 4.5 Interno di una cartella avente le funzioni di parsing

Una volta fatto si deve creare all'interno di tale percorso il file conf.php nel quale verranno inserite le espressioni regolari da utilizzare per quello specifico parser, utilizzato per quella specifica lingua per quella specifica tipologia di documento.

4.2.1 Struttura e analisi di un file conf

All'interno di file di conf.php, si trova un array denominato "rules" nel quale vengono inseriti gli elementi che il parser deve cercare all'interno del documento di testo insieme alle espressioni regolari che deve utilizzare per rilevare tali elementi.

```
$formula1 = preg_replace('/\s/', ' ', 'KUVENDI I REPUBLIKËS SË SHQIPËRI SË VENDOSI:');
$formula2 = preg_replace('/\s/', ' ', 'KUVENDI POPULLOR I REPUBLIKËS SË SHQIPËRISË VENDOSI:');

$rules = Array(
    "preambleInitList" => Array(
        "Në\s+mbështetje\s+të\s+(neneve|nenin|nenit|nenet|neni)",
    ),
    "preambleEndList" => Array (
        preg_replace('/(\w|Ë|:)/', '{1}\s*', $formula1),
        preg_replace('/(\w|Ë|:)/', '{1}\s*', $formula2),
    ),
    "conclusionsInitList" => Array(
        "K\s*R\s*Y\s*E\s*T\s*A\s*R(\s*(I|E))?",
        "Miratuar\s+(m|n)ë\s+datën?\s*d{1,2}(\.|-|\s|\V)\d{1,2}(\.|-|\s|\V)\d{4}\.?",
        "Shpallur\s+me\s+dekretin\s+nr.\s*d+",
    )
)
```

Figura 4.6 Contenuto del file conf.php del parser structure

Analizziamo uno dei file conf.php, in questo caso del parser structure, (figura 4.6) il quale permette di suddividere il documento nei quattro macro elementi, ossia: intestazione, preambolo, corpo e conclusioni. Al suo interno sono definiti tre elementi che il parser utilizza la ricerca:

- **preambleInitList** (lista di regular expression che segnano l'inizio del preambolo): Trovare l'inizio del preambolo permette allo stesso tempo di trovare la fine dell'intestazione. All'interno delle leggi albanesi il preambolo inizia con la formula iniziale traducibile in italiano come "a sostegno del/degli articolo/i" e ricercabile all'interno del documento albanese con la regex "Në\s+mbështetje\s+të\s+(neneve|nenin|nenit|nenet|neni)" la quale indica che le parole "Në mbështetje të" devono susseguirsi in questo ordine, separate da almeno un carattere del tipo whitespace, per poi essere ulteriormente seguite da una declinazione della parola neni (articolo).
- **preambleEndList** (lista di regular expression che segnano la fine del preambolo): Trovare la fine del preambolo permette allo stesso tempo di trovare l'inizio del corpo del documento. All'interno delle leggi albanesi il preambolo termina con una delle formule di emanazione "KUVENDI I REPUBLIKËS SË SHQIPËRI SË VENDOSI:" e "KUVENDI POPULLOR I REPUBLIKËS SË SHQIPËRISË VENDOSI:" in cui ciascun carattere può essere distanziato dall'altro da un whitespace. Le regex finali sarebbero dunque "K\s*U\s*V\s*E\s*N\s*D\s*I\s*...". Tuttavia, creare delle regular expression così lunghe avrebbe reso il codice poco mantenibile. Si è optato per l'utilizzo della funzione "preg_replace", prima per eliminare gli spazi bianchi dalla stringa e poi per aggiungere la tripla "\s*" dopo ogni carattere stampabile rimasto.

- **conclusionsInitList** (lista di regular expression che segnano l'inizio delle conclusioni): trovare l'inizio delle conclusioni permette di trovare la fine del corpo del documento. All'interno delle leggi albanesi il corpo termina con tre possibili formule: una indicante il capo del parlamento albanese, con tutte le possibili declinazioni del sostantivo, le cui lettere possono essere separate da un whitespace; la seconda possibile formula conclusiva ha come significato in italiano "Approvato in data", seguito dalla data di approvazione del documento; infine l'ultima formula conclusiva ha come significato italiano "Promulgato con il decreto n." seguita dal relativo numero.

Capitolo 5

Valutazioni

Terminata l'implementazione della funzione di markup automatico in lingua albanese, effettuiamo dei test su un campione di 10 documenti legislativi in modo da capire quali siano i punti di forza e i limiti dei parser e delle espressioni regolari che abbiamo implementato per il riconoscimento automatico degli elementi.

5.1 Uno sguardo d'insieme

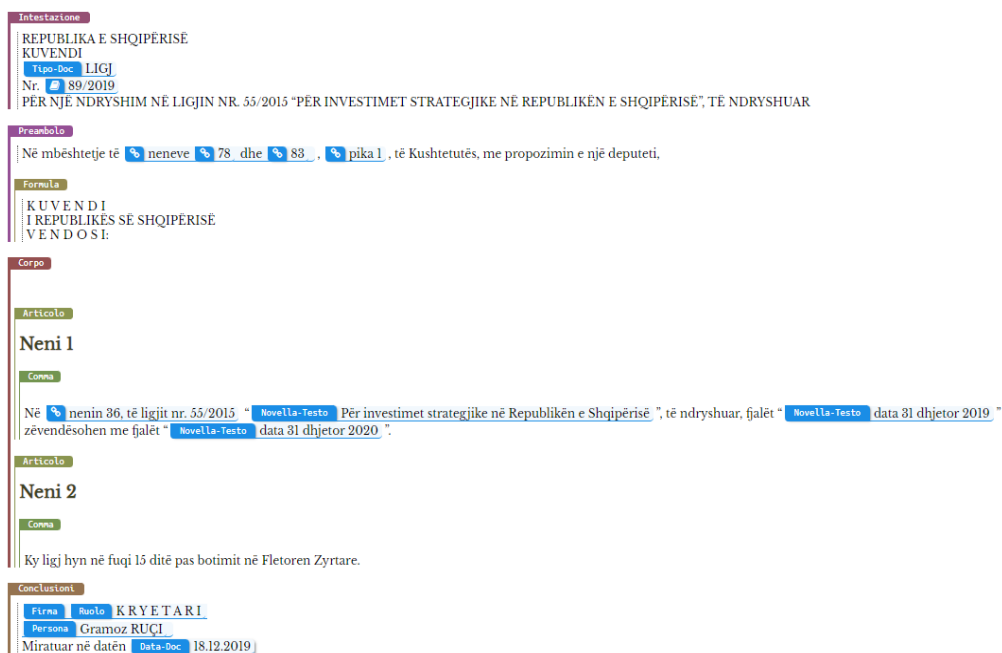


Figura 5.1 Documento la cui struttura e gli elementi vengono catturati dalla sola funzione di markup automatico

In tutti e 10 i documenti testati sono stati rilevati con successo i principali macroelementi che definiscono la struttura del testo, ossia: l'intestazione, il preambolo, il corpo e le conclusioni. I sottoelementi di queste sezioni vengono rilevati correttamente il 90% delle volte. La necessità di utilizzare la funzione di markup manuale diventa molto sporadica. Le eccezioni si riscontrano principalmente tra i riferimenti e le citazioni (tabella 5.1).

Nome elemento	n. elementi presenti	n. elementi rilevati correttamente	Tasso di successo
Preambolo	10	10	100,0%
Prefazione	10	10	100,0%
Corpo	10	10	100,0%
Conclusioni	10	10	100,0%
Tipo Doc.	10	10	100,0%
Numero Doc.	10	10	100,0%
Formula di emanazione	10	10	100,0%
Capitolo	4	4	100,0%
Articolo	93	93	100,0%
Paragrafo	102	98	96,08%
Lista	11	10	90,91%
Punti	46	42	91,30%
Riferimenti	104	75	72,12%
Quoted text	44	41	93,18%
Quoted structure	16	12	75,00%
Data Doc.	10	10	100,0%
Firma	10	10	100,0%

Tabella 5.1 Tasso degli elementi correttamente rilevati

5.2 Il problema dei riferimenti

All'interno dei 10 documenti utilizzati come test erano presenti 104 riferimenti di vario tipo (singoli, multipli, all'interno di un certo range) dei quali 75 (circa il 72%) rilevati correttamente. Essi rappresentano sicuramente la tipologia di elemento che più di tutti è difficile rilevare in ogni sua possibile sfumatura all'interno di un documento legislativo.

Oltre a poter essere classificati come singoli, multipli o all'interno di un certo range, i riferimenti possono avere ulteriori caratteristiche: possono riferirsi a parti interne o esterne al documento (nel cui caso si dovrà indicare con precisione il documento di provenienza), contenere date, oppure avere una struttura *una tantum*, che non viene più riscontrata altrove. Tipicamente i riferimenti sono scritti in maniera gerarchica: gli elementi più importanti sono scritti per primi, seguiti da quelli secondari tuttavia, vi possono essere delle eccezioni che il parser non riesce ad elaborare correttamente, in figura 5.2 un esempio di tale errore.

articolo Neni 24

pikës 4 , nenit 17

Percorso: act > body > article Neni 25 > paragraph 1. > mref > ref

Dati relativi al riferimento

Interno	dynamic
Nazionalità	Tipo
Sottotipo	Autore
Data	Numero
Linguaggio	Data versione
para_nenit.17	

Salva

Figura 5.2 Riferimento non elaborato correttamente

Non sempre le espressioni regolari riescono a cogliere una vasta gamma di possibili combinazioni di riferimenti e ciò non permette al parser di elaborare nel modo corretto tale classe di elementi con la stessa precisione che si vede con le altre tipologie.

Le espressioni regolari, per la loro natura di cercare pattern costanti all'interno della lingua naturale, non riescono a cogliere elementi la cui variabilità non sia prevedibile. Questo impedisce al parser di elaborare correttamente questa classe di elementi, diversa rispetto alle altre.

La soluzione migliore rimane dunque quella di creare sempre più espressioni regolari che riescano a cogliere un numero sempre più alto di elementi per questa specifica classe. Tuttavia risulta molto improbabile l'idea di riuscire a raccogliere correttamente in automatico ogni singolo elemento interno ad un documento senza che vi sia un'ulteriore revisione da parte dell'utente.

5.3 Il problema delle citazioni

Un ulteriore problema sorge quando, all'interno di un elemento *quoted structure* (citazione strutturata su più righe), vi sono annidati altri elementi i quali contengono al loro interno il carattere dei doppi apici (“ ”). Per esempio se all'interno di un elemento di tipo *quoted structure* ci sono altri elementi di tipo *quoted* (sia *text* che *structure*) essi non permettono al parser di distinguerli l'uno dall'altro in quanto esso non è stato pensato per gestire una tale casistica nella quale vi sono più elementi di questo tipo annidati tra loro.

Comma

Articolo

Neni 16/1

Comma

1. Në rastet kur në tokat bujqësore në administrim të ministrisë përgjegjëse për bujqësinë, objekt i qiradhënies, qiramarrësi kërkon të ndërtojë objekte për prodhimin, ruajtjen dhe përpunimin e produkteve bujqësore e blegtorale apo objekte në funksion të agroturizmit, për të cilat kërkohet leje ndërtimi dhe kategoria e resursit ndryshon nga

“tokë bujqësore” në “ **Novella-Testo** **truall** ’’, qiradhënësi mund të japë miratimin, nëse plotësohen kushtet e mëposhtme:

- Ndërtimi i objekteve të parashikuara në kërkesë të jetë në përputhje me legjislacionin dhe dokumentet e planifikimit të territorit;
- Sipërfaqja ndërtimore të shërbejë vetëm për objekte për prodhimin, ruajtjen dhe përpunimin e produkteve bujqësore e blegtorale, të tokës bujqësore, objekt i qiradhënies ose për ngritjen e objekteve ndërtimore në funksion të agroturizmit.

Figura 5.3 Citazioni annidate marcate in maniera scorretta

Tale errore è dovuto ad un limite del parser, il quale non è in grado di contare il numero di apici aperti durante l'attività di markup automatico e dunque permette la chiusura dell'elemento quoted non appena si presenta un qualunque tipo di doppio apice.

Capitolo 6

Conclusioni

In conclusione, si è mostrato il funzionamento del markup automatico nell'editor LIME ed in generale per il trattamento automatico dei documenti legislativi, e di come sia possibile poter ampliare il set delle lingue supportate.

All'inizio nel primo capitolo di questo testo è stata esposta una breve storia della digitalizzazione dei documenti legislativi e dei linguaggi di markup utilizzati per la descrizione. Successivamente, è stata fornita una dettagliata descrizione del linguaggio XML, sottolineando la sua importanza nel campo della codifica dei testi legislativi. È stato in seguito introdotto il formato Akoma Ntoso, spiegandone le sue caratteristiche peculiari e l'utilizzo nella standardizzazione dei documenti legislativi. Infine, è stata presentata una panoramica sull'editor online LIME, illustrando come tale strumento consenta la creazione, importazione, l'editing, la revisione e l'esportazione di testi legislativi in modo efficiente, ad un ipotetico nuovo utente che deve approcciarsi con tale strumento.

Inoltre, sono state spiegate le varie classi di macroelementi e di sottoelementi che si possono trovare all'interno di un documento legislativo, e con che caratteristiche e con quali peculiarità tali classi si presentano all'interno di una legge scritta in lingua albanese.

Il successivo capitolo fornisce la documentazione per aggiungere una nuova lingua alla customizzazione dell'editor, in particolare la struttura di un file contenente le espressioni regolari che permettono ai parser di ricercare all'interno del testo gli elementi desiderati, in modo che uno sviluppatore possa estendere e rendere tale funzionalità più efficiente.

Infine, è stato svolto un test su un *sample* di documenti in lingua albanese. La funzione di markup automatico è riuscita a processare la totalità dei macroelementi e la quasi totalità delle altre classi di elementi esistenti. La funzione, nel complesso, rende nella maggior parte dei casi sporadica la trattazione del documento con la funzione di markup manuale.

Tra le possibili migliorie da effettuare all'interno della funzione di markup automatico è necessaria una più corretta e precisa rilevazione dei riferimenti la quale può essere ottenuta tramite un'ulteriore aggiunta e affinamento delle espressioni regolari utilizzate dal parser per la rilevazione di tale classe di elementi. Un'ulteriore ritocco riguarda la rilevazione delle citazioni, al cui interno compaiono doppi apici senza che indichino l'inizio o il termine di tale citazione. Un dei casi possibili può essere la presenza di citazioni annidate all'interno di altre. Per questi casi, è richiesta l'evoluzione del parser che se ne occupa in modo che possa essere in grado di trattare citazioni annidate all'interno di altre citazioni.

Il parlamento albanese, il soggetto richiedente tali funzioni, ha espresso soddisfazione per lo strumento di lavoro e per le funzioni implementate, tanto da decidere di continuarne l'utilizzo e di richiedere di volta in volta l'implementazione di una sempre maggiore varietà di elementi da catturare e di funzionalità da aggiungere. L'ultima richiesta riguarda il trattamento automatico degli allegati all'interno dei documenti.

Si auspica dunque un ulteriore sviluppo delle funzioni di markup automatico implementate all'interno di tale editor per il trattamento di documenti giuridici e legislativi in questa e in altre lingue, e per ulteriori classi di documenti al fine di favorire una maggiore efficacia e precisione per l'elaborazione di una sempre maggiore varietà di testi.

Bibliografia

- [BF08] Biasiotti, M; Francesconi, E; Palmirani, M; Sartor, G., & Vitali, F. (2008). Legal informatics and management of legislative documents. Global Center for ICT in Parliament Working Paper, 2.
- [BB+99] Jon Bosak, Tim Bray, et al. Xml and the second-generation web. *Scientific American*, 280(5):89–93, 1999.
- [BPS97] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau. “Extensible Markup Language (XML).” *World Wide Web Journal* 2.4:27-66. 1997.
- [LS99] Håkon Wium Lie, Janne Saarela. "Multipurpose Web publishing using HTML, XML, and CSS." *Communications of the ACM* 42.10 (1999): 95-101.
- [PV12] Palmirani, M., & Vitali, F. (2012). Legislative XML: principles and technical tools. Inter-American Development Bank.
- [VZ07] Fabio Vitali and Flavio Zeni. Towards a country-independent data format: the akoma ntoso experience. In *Proceedings of the V legislative XML workshop*, pages 67–86. Florence, Italy: European Press Academic Publishing, 2007.
- [B99] Berners-Lee, Tim. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper San Francisco, 1999.
- [BPV11] Gioele Barabucci, Monica Palmirani, Fabio Vitali, and Luca Cervone. Long-term preservation of legal resources. In Kim Normann Andersen, Enrico Francesconi, Ake Grönlund, and Tom M. Engers, editors, *Electronic Government and the Information Systems Perspective*, volume 6866 of *Lecture Notes in Computer Science*, pages 78–93. Springer Berlin Heidelberg, 2011.

[BCP09] Gioele Barabucci, Luca Cervone, Monica Palmirani, Silvio Peroni, and Fabio Vitali "Multi-layer markup and ontological structures in Akoma Ntoso". In: International Workshop on AI Approaches to the Complexity of Legal Systems. Springer. 2009, pp. 133–149.

[PV11] Monica Palmirani and Fabio Vitali. "Akoma-Ntoso for legal documents". In: Legislative XML for the semantic Web. Springer, 2011, pp. 75–100.

[C13] Cervone L. (2013). Parametric editors for structured documents.

[T68] Thompson, Ken (1968). "Programming Techniques: Regular expression search algorithm". Communications of the ACM. 11 (6): 419–422.

[F12] Introducing Regular Expressions: Unraveling Regular Expressions, Step-by-Step, O'Reilly Media, Michael Fitzgerald, 2012, pp. 1-4.

[JPA68] Johnson, Walter L.; Porter, James H.; Ackley, Stephanie I.; Ross, Douglas T. (1968). "Automatic generation of efficient lexical processors using finite state techniques". Communications of the ACM. 11 (12): 805–813