Università degli Studi Roma Tre

Dipartimento di Ingegneria



Corso di Laurea Triennale in Ingegneria Informatica

XFP: Un Algoritmo per l'inferenza automatica della struttura di siti Web di grandi dimensioni basato   
sull'Analisi dei Punti Fissi

Relatore: Studente:

Dott. Valter Crescenzi Lukas Canciani Graziani

Anno Accademico 2018/2019

**Indice**

[Capitolo 1 Introduzione alla problematica - 1 -](#_Toc22026919)

[Capitolo 2 Tecniche risolutive - 3 -](#_Toc22026920)

[Capitolo 3 Strumenti utilizzati per la rappresentazione della soluzione - 13 -](#_Toc22026921)

[Capitolo 4 Soluzione - 18 -](#_Toc22026922)

[Capitolo 5 Criterio di selezione dell’ottimo - 25 -](#_Toc22026923)

[Capitolo 6 Esperimenti - 32 -](#_Toc22026924)

[Capitolo 7 Sviluppi futuri - 49 -](#_Toc22026925)

Lista figure

Figura 1- Esempio di DOM - 3 -

Figura 2- Esempio grafico di regola XPath - 5 -

Figura 3- Esempio grafico di un vettore di valori - 6 -

Figura 4- Interazioni di punto fisso della funzione coseno - 8 -

Figura 5- Ciclo di estrazione e generazione - 9 -

Figura 6- Esempio grafico di un punto fisso negli XPath - 10 -

Figura 7- Esempio di partizione - 14 -

Figura 8- Esempio di reticolo - 17 -

Figura 9- Precision e recall - 35 -

Figura 10- Grafico espressività 4 - 46 -

Figura 11- Grafico espressività 6 - 46 -

Figura 12- Grafico espressività 4 e 6 - 47 -

Figura 13- Grafico tempi di esecuzione - 47 -

**Lista tabelle**

Tabella 1- Dataset - 33 -

Tabella 2. Espressività 4 - Migliore Rank - 37 -

Tabella 3 - Espressività 4 - Migliori 2 Rank - 38 -

Tabella 4 - Espressività 4 - Migliori 3 Rank - 40 -

Tabella 5 - Espressività 6 - Migliore Rank - 41 -

Tabella 6 - Espressività 6 - Migliori 2 Rank - 43 -

Tabella 7 - Espressività 6 - Migliori 3 Rank - 44 -

# Capitolo 1 Introduzione alla problematica

Con l’avvento di internet le informazioni sono sempre più rapidamente fruibili ed accessibili quasi costantemente agli utenti, tuttavia anche la mole di informazioni presenti sul web è in rapido aumento, rendendo più complessa la ricerca delle informazioni desiderate.

Nascono per ovviare a questa problematica diversi algoritmi di estrazione dati, che consentono tramite alcune “istruzioni” preventivamente inserite dall’utente di isolare ed estrarre solo alcune informazioni da una determinata pagina web semplificando e rendendo estremamente più veloce la reperibilità delle informazioni cercate.

Si può ad esempio pensare ad un software per il confronto dei prezzi tra più siti di e-commerce. Una volta identificato un prodotto il software sarà in grado di effettuare la ricerca sui siti di e-commerce preventivamente configurati e di estrarre, tramite le istruzioni specifiche di ogni sito, solamente il prezzo del prodotto ricercato.

* 1. **Obiettivo**Lo scopo che l’algoritmo sviluppato durante lo svolgimento dell’attività progettuale di laboratorio si pone è quello di fornire un supporto per un’attività di estrazione “intelligente”, che sia dunque in grado di riconoscere ed estrarre i dati rilevanti in modo automatico senza la necessità di avere delle istruzioni per ogni sito.

’algoritmo si occupa quindi di effettuare un’analisi preventiva di un sito web, tramite tecniche e strumenti che verranno esposti nei capitoli seguenti, suddividendone le pagine in insiemi di pagine tra loro simili, dal punto di vista strutturale e di navigazione, ottenendo quindi, automaticamente, una descrizione della struttura del sito web e identificando alcune caratteristiche peculiari di questi sottoinsiemi.

* 1. **Struttura del contenuto**La tesi è articolata in sei capitoli, di cui seguirà una breve descrizione:

Il primo capitolo contiene una breve introduzione all’obbiettivo della tesi e ad una sua possibile applicazione pratica

Il secondo capitolo si sofferma sulle tecniche utilizzate per la risoluzione

Nel terzo capitolo vengono presentati gli strumenti utilizzati per la rappresentazione della soluzione

Nel quarto capitolo viene presentata la soluzione ottenuta dall’algoritmo

Il quinto capitolo si occupa di illustrare il criterio per la scelta della soluzione

Nel sesto capitolo vengono mostrati i resoconti degli esperimenti effettuati sull’algoritmo

Nel settimo ed ultimo capitolo vengono illustrati alcuni possibili sviluppi futuri

# Capitolo 2 Tecniche risolutive

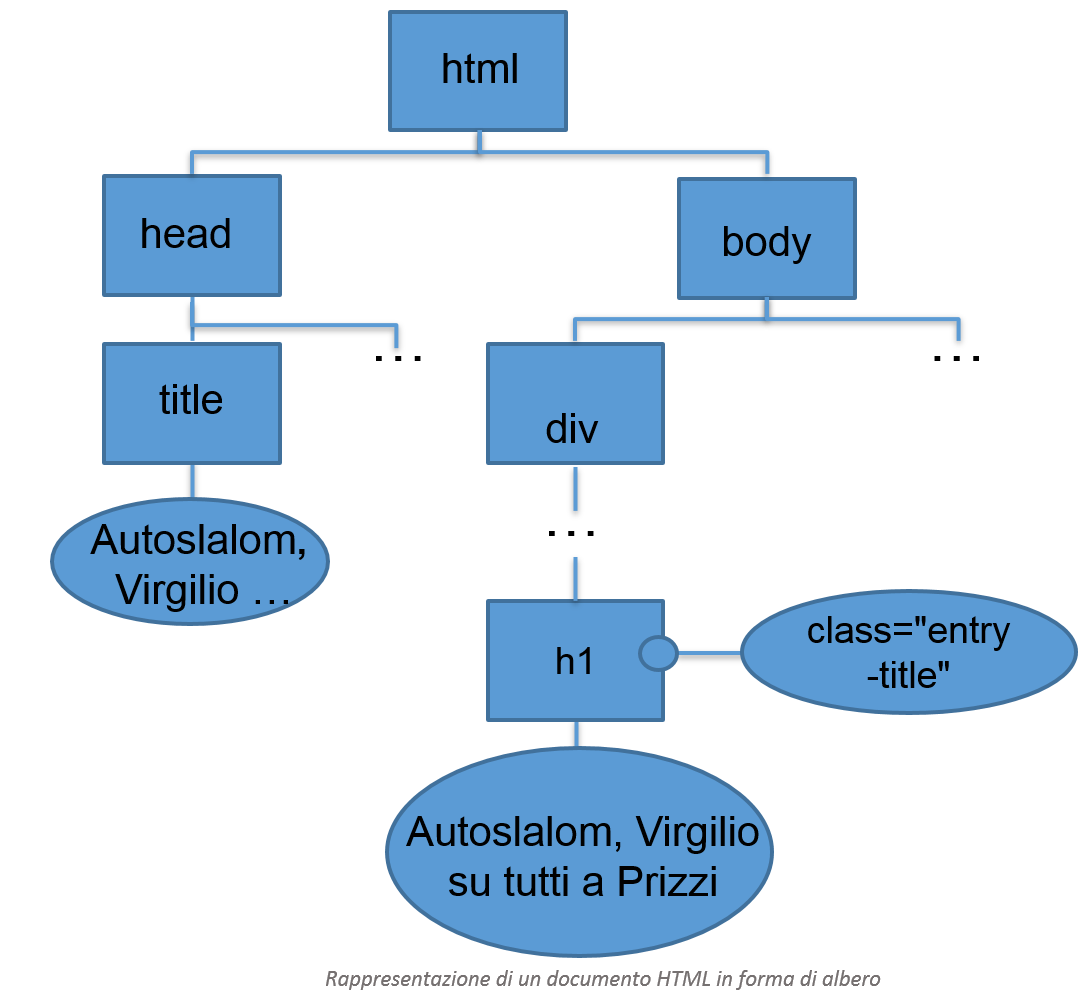
Le principali tecniche utilizzate nell’analisi di una o più pagine web sono prevalentemente due, un linguaggio di interrogazione XML denominato XPath ed un concetto matematico, il punto fisso.

**2.1 XPath**

Nel corrente capitolo verrà trattato brevemente il concetto di XPath ai soli fini di fornire una comprensione basilare e poco specifica, sufficiente per la comprensione dei concetti principali del restante contenuto della tesi. Per un ulteriore approfondimento sull’argomento è disponibile la documentazione ufficiale in lingua inglese sul sito del W3C, ente sviluppatore degli XPath. È inoltre presente un riassunto in lingua italiana dei principali punti di tale documentazione nell’appendice.

XPath, XML Path language, è un linguaggio di ricerca che consente l’identificazione di uno o più nodi di un file XML nella sua rappresentazione DOM, ovvero sotto forma di albero.

Figura 1- Esempio di DOM



Il linguaggio è composto da espressioni, dette anche regole, composte da più passi. Ognuno dei passi pone delle condizioni e restituisce solamente i nodi che le rispettano. I nodi restituiti da ogni passo verranno quindi sottoposti alla verifica delle condizioni del passo successivo. L’insieme di nodi restituiti dalla regola corrisponde a quello eventualmente restituito dall’ultimo passo della regola stessa.

**2.1.1 Esempi di XPath**

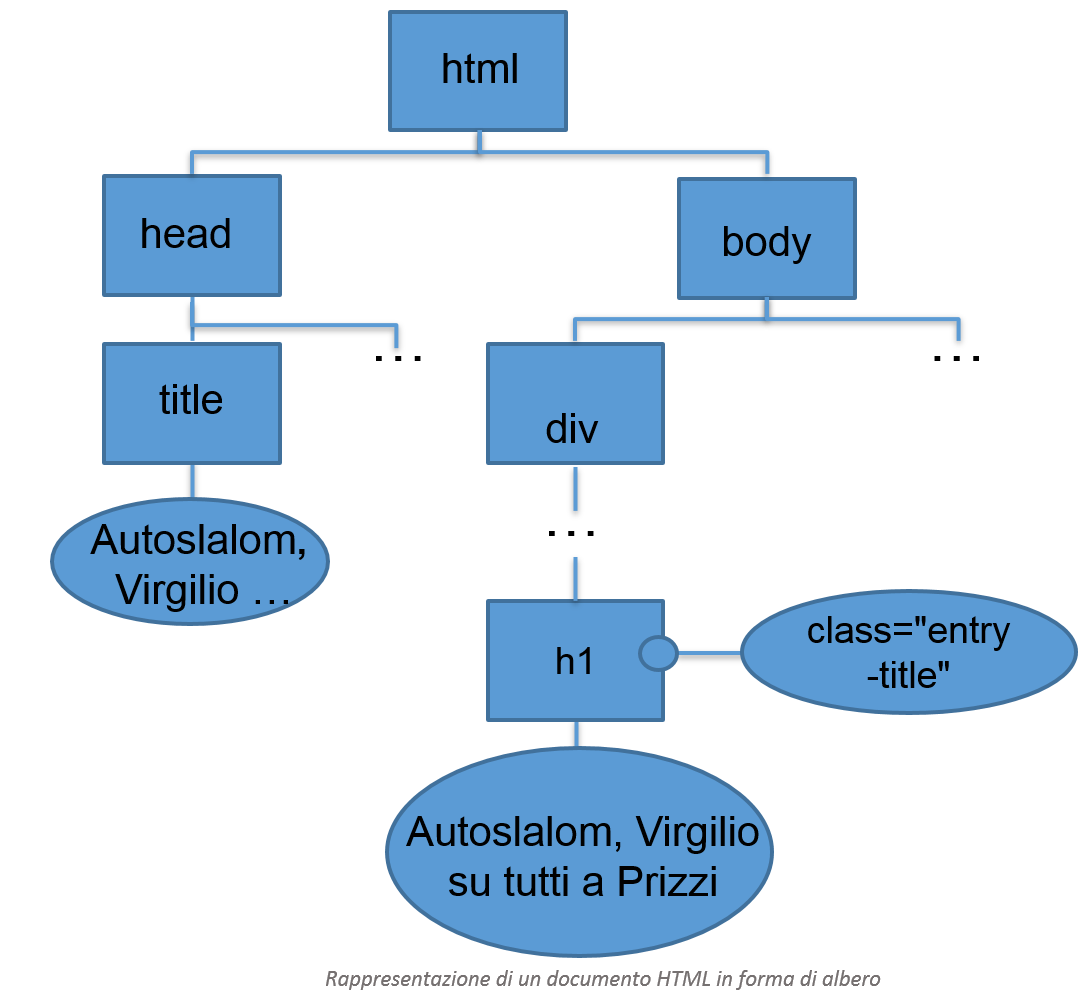
Si ha ad esempio che una regola XPath *“//a[@href=”link.html”]”* (equivalentea “*descendant-or self::a[@href=”link.html”]”)* applicata ad un file HTML identifica un qualsiasi nodo, discendente della radice o la radice stessa, che sia di tipo *a* e che contenga un attributo *href* con valore pari a “*link.html”*.

Un regola *“descendant::ul/child::li[last()]”* identifica invece ogni ultimo elemento di una qualsiasi lista non ordinata discendente della radice.

**2.1.2 Esempio Grafico**

Si presenta quindi un esempio di regola XPath identificando graficamente il valore che estrae, sia a livello di visualizzazione HTML sia a livello di DOM.

Figura 2- Esempio grafico di regola XPath



//h1[@class="entry-title"]

**2.1.3 Vettore di valori**

Una regola XPath identifica nodi da cui estrarre valori se applicata ad un documento HTML. Se una stessa regola viene applicata ad un insieme di documenti HTML per cui è valida si estrarrà uno o più valori, che possono comunque essere nulli, per ognuno dei documenti. Tali valori verranno poi raggruppati in un vettore di valori.

Figura 3- Esempio grafico di un vettore di valori

//h1[@class="entry-title"]



Vettore di valori

**Autoslalom, Virgilio su tutti a Prizzi**

**Perfezionata l’iscrizione in serie C del Trapani Calcio**

**Gossa perdita idrica, a Pizzolungo disagi nella erogazione**

**2.1.4 Estrazione di regole XPath**

È possibile estrarre automaticamente, tramite uno specifico algoritmo detto Estrattore(E),a partire da un documento XML un insieme di regole che applicate a tale documento estraggano dei valori. Ogni regola estratta identificherà quindi uno o più elementi della pagina sorgente. Il particolare algoritmo utilizzato è reso configurabile consentendo di impostare a che livello di profondità si vuole scendere all’interno dell’albero per estrarre regole.

**2.1.5 Generazione di regole XPath**

Si può inoltre generare delle regole XPath a partire da uno specifico vettore di valori, utilizzando un algoritmo detto Generatore(G). Tale algoritmo prendere in input un vettore di valori estratti da un insieme di documenti HTML e restituisce in output un insieme di regole XPath che applicate allo stesso insieme di documenti consenta di estrarre lo stesso vettore di valori. L’algoritmo Generatore inoltre utilizza un frammento XPath configurabile che stabilisce il criterio secondo cui generare queste regole e stabilisce quali regole siano accettabili. Utilizzare un frammento XPath si dimostra indispensabile per limitare il numero di regole che risulterebbe altrimenti troppo elevato.

**2.2 Punto fisso**

“In matematica, un p. f. per una funzione definita su un insieme A è un elemento .” [[1]](#footnote-1)

Si definisce quindi, in matematica, come punto fisso per una funzione un punto la cui immagine corrisponde con il punto stesso.

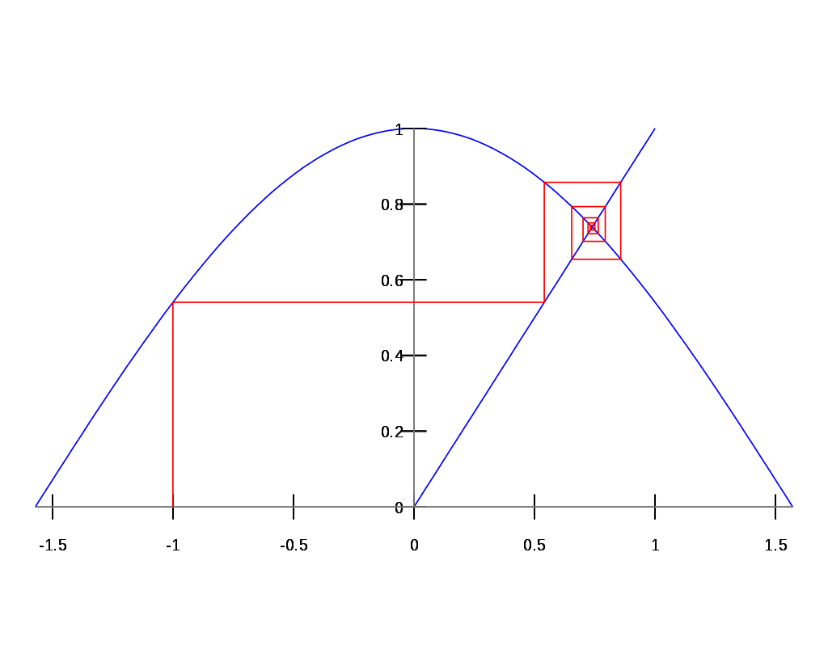
Ovvero è considerato punto fisso di una funzione un valore che resta immutato all’applicazione di tale funzione.

Ponendo un esempio pratico si avrà che per la funzione il punto x=2 sarà un punto fisso poiché

Alcune funzioni hanno poi la caratteristica di “convergere” ad un punto fisso //*detto punto fisso convergente o aggregante* se iterate ripetutamente a partire da un qualsiasi valore, o da un valore contenuto in un determinato intervallo. Si ha quindi che *f(f(f(...(f(x)...))) = x.*

Un esempio è la funzione coseno: iterando a partire da un qualsiasi numero reale la funzione coseno si vedrà come il risultato converge verso 0.739085133, unico punto fisso della funzione coseno, come visibile nella Figura 4. *[[2]](#footnote-2)***2.2.1 Punto fisso nell’ambito degli XPath**

Figura 4- Interazioni di punto fisso della funzione coseno



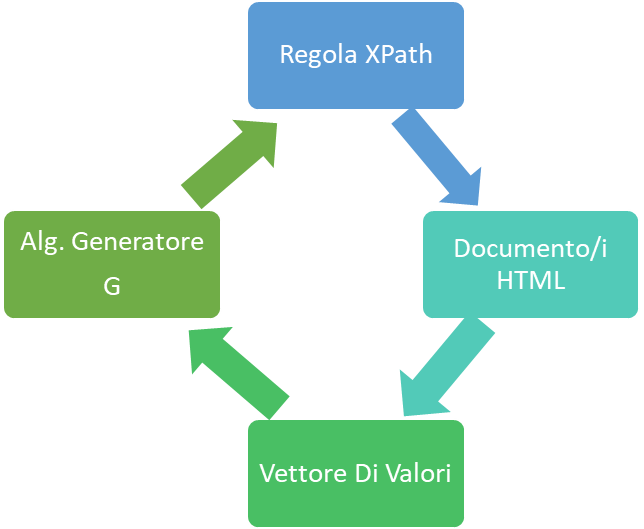
Viene quindi introdotto il concetto di punto fisso nell’ambito delle regole XPath e dei file XML.

Come visto in precedenza una regola XPath identifica uno o più valori, anche nulli, in un file HTML o un vettore di valori in un insieme di file.

Si definisce quindi punto fisso per un file HTML una regola XPath che si autorigeneri, ovvero che ricompaia nell’insieme di regole generate a partire dai valori che estrae.

Quindi per verificare se una regola generata su di un file XML sia un punto fisso per quello stesso file occorre “estrarre” i valori dei nodi identificati dalla regola stessa e successivamente generare, utilizzando l’algoritmo Generatore, delle nuove regole XPath a partire dai valori identificati. Qualora tra le nuove regole generate vi fosse quella di partenza allora è possibile considerare la regola originaria come punto fisso di quel documento o di quell’insieme di documenti.

Figura 5- Ciclo di estrazione e generazione



Applicata a

Esrae

Frammento XPath

Genera

**2.2.2 Esempio grafico di punto fisso negli XPath**

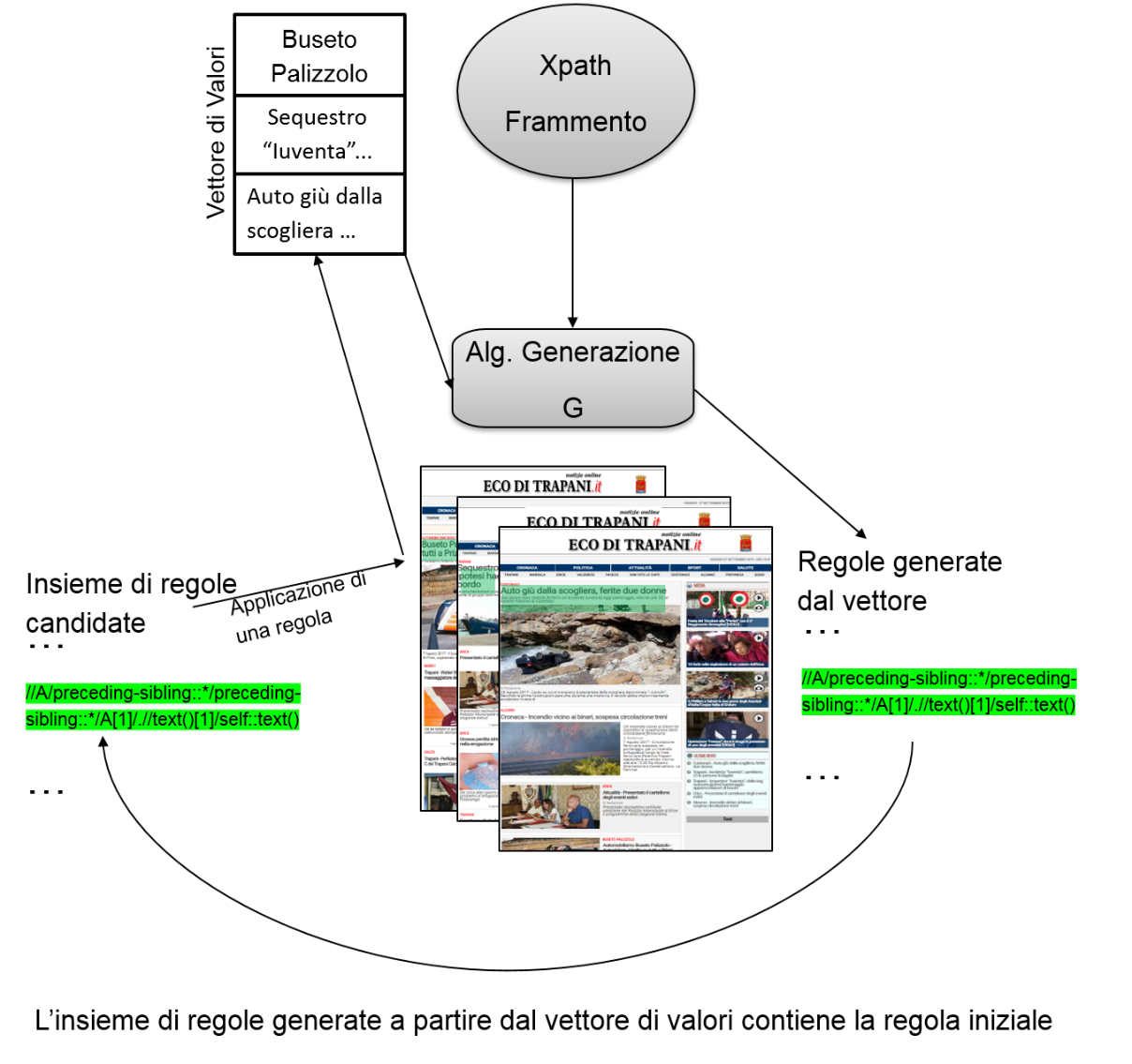


Figura 6- Esempio grafico di un punto fisso negli XPath

*di*

**2.2.3 Classificazione dei punti fissi negli XPath**

Un punto fisso di un insieme di file XML può essere classificato in base ad alcune caratteristiche:

1. Costante o variabile
2. Opzionale o non opzionale
3. Navigazionale o di dati.
4. Un punto fisso si dice costante quando i valori di tutti i nodi estratti dalla regola XPath sui vari file HTML, ovvero quando tutti i valori del vettore estratto, sono uguali e si dice viceversa variabile quando i valori estratti differiscono tra loro. Applicando tale differenza al contesto di questa tesi tale classificazione coincide con una notevole differenza di rilevanza di un punto fisso. Nell’analisi di un insieme di pagine HTML la presenza di un punto fisso identifica un elemento presente in ogni pagina, se il punto fisso è costante tale elemento conterrà in ognuna delle pagine lo stesso valore e pertanto si avrà, con alta probabilità, che tale punto fisso identifichi un elemento della struttura dell’intero sito o dell’insieme di pagine analizzato, e che pertanto il suo contenuto non sia rilevante nell’analisi del contenuto di una singola pagina. Se il punto fisso invece è variabile si avrà che l’elemento, presente in ogni pagina, avrà un valore sempre diverso e specifico per ogni pagina assumendo con alta probabilità un importante ruolo nell’analisi del contenuto della pagina. Si nota dunque come i punti fissi variabili abbiano una rilevanza più alta rispetto ai punti fissi costanti nell’ambito dell’analisi dei contenuti di un insieme di pagine HTML.
5. Una regola XPath per essere punto fisso di un insieme di pagine HTML non deve necessariamente avere almeno un valore non nullo per ognuna delle pagine, è possibile che in una o più delle pagine la regola non estragga alcun valore dai nodi identificati ed in tale caso il punto fisso viene definito opzionale. Viceversa, una regola che estragga almeno un valore per ognuna delle pagine HTML di cui è punto fisso si dice non opzionale. Una regola non opzionale ha un’influenza decisamente più alta di una regola opzionale poiché è relativa a tutto l’insieme considerato e non solo ad un suo sottoinsieme.
6. È possibile introdurre un ulteriore classificazione basata sulla tipologia del contenuto degli elementi estratti da un punto fisso, un punto fisso si dice navigazionale qualora i valori estratti dai nodi identificati siano dei riferimenti ad altri documenti HTML e si dice invece di dati nell’eventualità non contengano collegamenti ad altre pagine ma dati in una qualsiasi forma.

# Capitolo 3 Strumenti utilizzati per la rappresentazione della soluzione

Al fine di rappresentare la soluzione al problema, di conservare i dati ottenuti durante l’analisi e di confrontarli con maggiore semplicità si è resa necessaria la creazione di alcune strutture. Si considera per le seguenti definizioni che l’analisi venga effettuata su un numero finito di pagine di un sito web, detto insieme iniziale.

**3.1 Classe di pagine**

In matematica si definisce una classe come:

“collezione di oggetti astratti, definiti da una specifica proprietà.”[[3]](#footnote-3)

Dopo aver selezionato le pagine web del sito su cui effettuare l’analisi viene effettuata su ognuna di esse un’estrazione di svariate regole XPath. Le pagine vengo poi raggruppate in insiemi di pagine che condividono una o più regole. Tali insiemi prendono il nome di classi di pagine. Ogni classe di pagine è quindi caratterizzata da un insieme di pagine e da un insieme di regole XPath che identificano nodi su quelle pagine.

Si avrà quindi che una classe di pagine è un insieme di pagine di uno stesso sito che condividono, almeno in parte, la stessa struttura presentando però contenuti differenti.

Ognuna delle regole XPath di una classe di pagine estrae un vettore di valori dalle pagine della classe.

**3.2 XPath caratteristico**

Si definisce una regola XPath come caratteristica di una classe di pagine qualora tale regola identifichi dei nodi se e solo se applicata alle pagine di quella specifica classe di pagine. Tale regola deve quindi necessariamente identificare almeno un nodo per ognuna delle pagine che compongono la classe di pagine e non deve identificare nodi per nessuna delle altre pagine.

**3.3 Partizione**

“In matematica, nella teoria degli insiemi, p. in classi di un insieme, la sua suddivisione in sottoinsiemi, a due a due disgiunti, che abbiano come unione l’insieme stesso.”[[4]](#footnote-4)

Si intende, in questo contesto, per partizione un insieme di classi di pagine tali che ogni pagina dell’insieme iniziale sia presente in una ed una sola classe di pagine. Si otterrà quindi una “mappatura” completa dell’insieme di interesse in cui l’insieme delle pagine delle classi di pagine darà l’intero insieme iniziale senza ripetizioni di pagine.

Figura 7- Esempio di partizione



7

4

12

6

L’insieme delle partizioni viene attualmente generato a partire dalle classi di pagine precedentemente generate. Vengono quindi create tutte le partizioni possibili, ovvero tutti gli insiemi di classi di pagine che risultano partizioni. Tale soluzione è dispendiosa in termini di tempi di esecuzione e aumenta più che proporzionalmente all’aumentare delle pagine dell’insieme iniziale. Il numero massimo di partizioni a cui si può arrivare con un insieme iniziale di *n* pagine è dato dal numero di Bell.

Poiché non vengono generate tutte le classi di pagine possibili, ma solamente quelle le cui pagine sono accomunate da una stessa struttura, almeno parziale, il numero generato di partizioni risulta in tutti i casi studiati di molto inferiore. Sarebbe tuttavia possibile arrivare con l’algoritmo attuale a tale numero di partizioni qualora ogni possibile combinazione di pagine avesse una parziale struttura condivisa solamente da quella combinazione stessa.

Una soluzione minore e temporanea che è stata applicata alla generazione consiste nel rimuovere dall’algoritmo di generazione le classi di pagine “singleton”, ovvero le classi di pagine contenenti un'unica pagina HTML, e solo nel caso la generazione di una partizione si concludesse senza aver ricoperto l’intero sito andare ad inserire le pagine mancanti come singleton, qualora fossero presenti le classi singleton corrispondenti a tali pagine.

Questa soluzione è solamente temporanea ed è sicuramente necessario modificare l’algoritmo di generazione delle partizioni per poter eseguire l’algoritmo su un intero sito reale, tuttavia essendo quello sviluppato un prototipo la priorità è stata assegnata alla completezza del reticolo generato piuttosto che ai tempi di esecuzione.

**3.3.1 Criterio di ordinamento tra partizioni**

Al fine di poter confrontare le partizioni generate si è reso necessario introdurre un criterio di ordinamento tra le partizioni in funzione delle classi di pagine che contengono.

Si introduce quindi il concetto di raffinamento per il confronto tra classi di pagine definendo una classe di pagine raffinamento di un’altra classe di pagine tale che se e solo se contiene tutte le pagine che pongono ed altre. Si ha quindi che

Si può quindi definire il raffinamento tra partizioni in funzione delle loro classi di pagine: Si ha che una partizione si dice raffinamento di una partizione tale che , qualora ogni classe di pagine di sia contenuta in o sia raffinata da una classe di pagine di .

**3.4 Reticolo**

Per la rappresentazione dei risultati parziali, e per conservarne in maniera efficiente le relazioni, viene introdotto il principio di reticolo, o *lattice* in inglese.

“reticolo in algebra, insieme R dotato di un ordinamento parziale (largo) tale che per ogni coppia di elementi a e b di R, sono definiti il loro → minimo comune maggiorante (denotato con a ∨ b) e il loro → massimo comune minorante (denotato con a ∧ b).”[[5]](#footnote-5)

Un reticolo è quindi un particolare insieme parzialmente ordinato secondo un dato criterio. Tale criterio nel caso delle partizioni di classi di pagine trattate corrisponde al raffinamento. Si ottiene quindi una struttura reticolare che parte da partizioni poco “raffinate” progredendo con partizioni sempre più fine.

Figura 8- Esempio di reticolo



*di*

# Capitolo 4 Soluzione

Occorre quindi definire la forma dei risultati dell’analisi che si effettua.

**4.1 Struttura pregressa**

Tramite le tecniche e gli strumenti rappresentativi visti nei capitoli precedenti è possibile ottenere un reticolo “completo” comprendente ogni partizione che è possibile ottenere sulla base delle classi di pagine generate.

Il primo passo che viene eseguito è una estrazione di regole XPath su ognuna delle pagine HTML presenti in input. Tale estrazione è stata delegata ad un algoritmo fornito e sviluppato dal Professor Crescenzi. Vengono poi eliminate le regole che risultano essere equivalenti, sia in termini di regola stessa, che di valori estratti sulla pagina dalla regola.

*Set<XPath> rulesGeneration(Set<Page> pages){*

*Map<Page, Set<String>> p2x*

*Set<String> diffRules*

*for (page : pages){*

*Set<String> rules = inferRules(page)*

*diffRules.addall(rules)*

*p2x.put(page,rules)*

*}*

*Set<XPath> xpaths*

*for (rule : diffRules){*

*Set<Page> rulePages*

*for(page : p2x.key()){*

*if(p2x.get(page).contain(rule))*

*rulePages.add(page)*

*}*

*xpath = new XPath(rule,rulePages)*

*xpaths.add(xpath)*

*for(page : rulePages)*

*page.addXPath(xpath)*

*}*

*return xpaths*

*}*

Si procede quindi al raggruppamento delle pagine nelle varie classi di pagine. Tali classi di pagine raggruppano pagine che condividono regole XPath che sono comuni solo a quel preciso gruppo e a nessun altro. Ognuna di tali regole, specifica per una singola classe di pagine, è detta XPath caratteristico.

*Set<PageClass> groupPagesByXPaths(Set<XPath> xpaths){*

*Set<PageClass> pageClasses*

*for(xpath : xpaths){*

*Set<XPath> pageClassRules*

*pageClassRules.add(xpath)*

*for(xpath2 : xpaths){*

*if (xpath != xpath2 && checkSamePages(xpath,xpath2)){*

*pageClassRules.add(xpath2)*

*}*

*}*

*pageClass = new PageClass(xpath.getPages(), pageClassRules)*

*if(!pageClasses.containsXPathsSet(pageClass)*

*pageClasses.add(pageclass)*

*}*

*return pageClasses*

*}*

Vengono poi formate tutte le partizioni che è possibile generare a partire dalle classi di pagine ottenute.

*private static Lattice generatePartitions(Set<PageClass> pageClasses, int MAX\_PAGES) {*

*Set<Partition> partitions*

*Set<PageClass> toCheck*

*Set<PageClass> singles*

*for(PageClass pageClass : pageClasses) {*

*if(pageClass.getPages().size() == 1) {*

*singles.add(pageClass)*

*}else {*

*toCheck.add(pageClass)*

*}*

*}*

*for (PageClass pageClass : pageClasses) {*

*if(pageClass.getPages().size() != 1) {*

*toCheck.remove(pageClass)*

*Set<PageClass> classes*

*classes.add(pageClass) partitions.addAll(allPossiblePartitions(classes,toCheck,singles,*

*+ pageClass.getPages().size(),MAX\_PAGES))*

*}*

*}*

*Partition.reorderPartitions(partitions)*

*return new lattice(partitions)*

*}*

*private static Set<Partition> allPossiblePartitions(Set<PageClass> currentClasses ,*

*+ Set<PageClass> toCheck, Set<PageClass> singles, int pageCount ,int max){*

*final Set<Partition> partitions*

*boolean last = true*

*if (pageCount == max) {*

*final Partition partition = new Partition(currentClasses)*

*partitions.add(partition)*

*last = false*

*}*

*else if(!toCheck.isEmpty()) {*

*Set<PageClass> newToCheck*

*newToCheck.addAll(toCheck)*

*for (PageClass PClass : toCheck) {*

*newToCheck.remove(PClass)*

*if ((pageCount+PClass.getPages().size()<=max)*

*+&&(!PClass.hasSamePagesAs(currentClasses))){*

*Set<PageClass> newClasses*

*newClasses.addAll(currentClasses)*

*newClasses.add(PClass)*

*Set<Partition> partials = allPossiblePartitions(newClasses,*

*+ newToCheck,singles,(pageCount+PClass.getPages().size()),max)*

*if(!partials.isEmpty()) {*

*last = false*

*}*

*partitions.addAll(partials)*

*}*

*}*

*}*

*if(last && singles!=null) {*

*for(PageClass single : singles) {*

*if(!single.hasSamePagesAs(currentClasses)) {*

*currentClasses.add(single)*

*pageCount++*

*}*

*}*

*if(pageCount == max) {*

*final Partition partition = new Partition(currentClasses)*

*partitions.add(partition)*

*}*

*}*

*}*

Si procede infine a verificare le relazioni di raffinamento presenti tra le partizioni ed a memorizzare il tutto in un reticolo.

*public Lattice(Set<Partition> partitions) {*

*this.partitions=partitions*

*this.isRefinedBy = new Map<Partition,Set<Partition>>()*

*this.isRefinementOf=new Map<Partition,Set<Partition>>()*

*this.GenerateRefinements()*

*}*

*private void GenerateRefinements() {*

*for(Partition p1 : this.partitions) {*

*for(Partition p2 : this.partitions) {*

*if(p1.isRefinementOf(p2)) {*

*this.addRefinement(p1,p2)*

*}*

*}*

*}*

*}*

*private void addRefinement(Partition finer, Partition coarser) {*

*if (this.isRefinementOf.containsKey(finer)) {*

*Set<Partition> refined = this.isRefinementOf.get(finer)*

*refined.add(coarser)*

*this.isRefinementOf.put(finer, refined)*

*}*

*else {*

*Set<Partition> refined = new Set<Partition>()*

*refined.add(coarser)*

*this.isRefinementOf.put(finer, refined)*

*}*

*if (this.isRefinedBy.containsKey(coarser)) {*

*Set<Partition> refinements = this.isRefinedBy.get(coarser)*

*refinements.add(finer)*

*this.getIsRefinedBy().put(coarser, refinements)*

*}*

*else {*

*Set<Partition> refinements = new Set<Partition>()*

*refinements.add(finer)*

*this.getIsRefinedBy().put(coarser, refinements)*

*}*

*}*

***Partition***

*public boolean isRefinementOf(Partition i) {*

*boolean isRefinement=true*

*//creo 2 mappe che contengono le pagine divise come sono divise nelle partizioni in input*

*p1 = new Map<Set<Page>,String> ()*

*p2 = new Map<Set<Page>,String ()*

*for(PageClass class : this.getPageClasses()) {*

*pages = new Set<Page>()*

*pages.addAll(class.getPages())*

*p1.put(pages, class.getId())*

*}*

*for(PageClass class2 : i.getPageClasses()) {*

*pages2 = new Set<Page>();*

*pages2.addAll(class2.getPages())*

*p2.put(pages2, class2.getId())*

*}*

*//itero su tutti i raggruppamenti di pagine della partizione "principale" e per ognuno di questi creo un insieme di IDPAGINA per effettuare i controlli*

*if(!this.samePartition(i)) {*

*for(Set<Page> pages2 : p2.keySet()) {*

*idsToCheck2 = new Set<String>()*

*for(Page temp : pages2) {*

*idsToCheck2.add(temp.getId())*

*}*

*//itero su tutti i raggruppamenti della partizione che vogliamo sapere se è raffinamento della principale e creo un insieme di IDPAGINA*

*for(Set<Page> pages1 : p1.keySet()) {*

*idsToCheck1 = new Set<String>()*

*for(Page temp : pages1) {*

*idsToCheck1.add(temp.getId())*

*}*

*//per ogni IDPAGINA del raggruppamento principale, il corrispettivo IDPAGINA della partizione da controllare e il raggruppamento a cui appartiene devono rispettare una condizione*

*for(String id : idsToCheck2) {*

*//se un raggruppamento della partizione da controllare contiene un id della partizione principale quest'ultima deve contenere TUTTI gli id del raggruppamento della partizione da controllare*

*if(idsToCheck1.contains(id) &&*

*+ !idsToCheck2.containsAll(idsToCheck1)) {*

*return isRefinement=false*

*}*

*}*

*}*

*}*

*}*

*else isRefinement=false*

*return isRefinement*

*}*

**4.2 Rappresentazione della soluzione**

Poiché lo scopo finale di questa analisi preventiva consiste nel fornire una rappresentazione della struttura del sito analizzato e nello specifico una suddivisione delle pagine HTML contenute nel dataset da analizzare la struttura di dati più idonea alla rappresentazione di tali informazioni è una partizione. Occorre quindi, preso l’intero reticolo, ordinato in termini di raffinamento stabilire quale delle partizioni individuate sia quella “ottima” e dunque quanto sia “fine” la soluzione cercata.

# Capitolo 5 Criterio di selezione dell’ottimo

Al fine di selezione la partizione ottima occorre effettuare un’analisi dei punti fissi delle varie classi di pagine che compongono ogni possibile partizione candidata.

**5.1 Ricerca dei Punti Fissi**

Il funzionamento della ricerca di punti fissi viene effettuata in due momenti distinti per isolare punti fissi di dati e punti fissi navigazionali, tuttavia le differenze tra i due procedimenti sono minime. La ricerca dei punti fissi restituisce punti fissi costanti e variabili e opzionali e non opzionali identificandoli e restituendone anche la classificazione.

La ricerca di punti fissi è effettuata su ogni classe di pagine singolarmente.

**5.1.1 Estrazione delle regole**

Il primo passo per effettuare la ricerca dei punti fissi è quello di estrarre le regole XPath candidate ad essere punti fissi. L’estrazione è stata delegata ad un algoritmo precedentemente sviluppato dal Prof. Crescenzi che viene utilizzato come scatola nera.

L’estrattore è configurabile, sia per quanto riguarda la “sensibilità” dell’estrazione, al pari del frammento XPath utilizzato dall’algoritmo Generatore, che si vuole effettuare, ovvero quanto in profondità nell’albero deve essere effettuata la ricerca di regole candidate, sia per quanto riguarda il tipo di punto fisso che si vuole estrarre. L’estrazione è differenziata tra ricerca di punti fissi di dati e punti fissi navigazionali poiché chiaramente la tipologia di nodi che si sta ricercando è nettamente differente: Nel caso di punti fissi navigazionali vengono cercati collegamenti ad altre pagine HTML, identificabili da un elemento “*a”* contenente un attributo *“href”*, mentre nella ricerca di punti fissi di dati vengono cercati nodi contenti appunto dati, informazioni di qualsiasi tipologia.

**5.1.2 Estrazione dei valori**

Per ognuna delle regole XPath candidate ad essere punto fisso vengono estratti i valori contenuti nei nodi che il percorso identifica. L’estrazione consente di classificare l’eventuale punto fisso di una determinata regola in funzione dei valori presenti: Se i valori estratti risultassero tutti uguali l’eventuale punto fisso sarebbe costante, viceversa se i valori differissero tra loro sarebbe variabile, cosi come, se per una delle pagine facenti parte della classe di pagine in analisi non fosse presente neanche un valore si avrebbe un punto fisso opzionale.

Le regole vengono poi raggruppate per valori estratti e quindi si raccolgono in un insieme tutte le regole XPath che portano, per ogni pagina, all’estrazione degli stessi valori.

**5.1.3 Verifica delle regole XPath candidate**

Ognuna delle regole XPath candidate viene quindi sottoposta singolarmente ad una verifica per stabilire se identifica o meno un punto fisso. Viene preventivamente effettuata una verifica sui soli valori estratti e qualora ci si trovasse in presenza di un vettore di valori *singleton*, ovvero un vettore di valori, estratti dalla regola XPath, che presenta complessivamente, su tutte le pagine che compongo la classe di pagine attualmente analizzata, un unico valore non nullo, esso viene immediatamente scartato, poiché poco rilevante. In caso di un vettore non *singleton* si procede con l’analisi e si generano delle nuove regole XPath a partire dai valori estratti. Si ottiene quindi un nuovo insieme di regole XPath, indipendenti dalle regole che hanno permesso l’estrazione iniziale del vettore di valori, i cui cammini portano però allo stesso insieme di valori. Qualora l’intersezione dei due insiemi di regole XPath, quello iniziale e quello generato a partire dai valori estratti, fosse non vuoto e dunque se vi fosse almeno una regola XPath in comune, si potrebbe ritenere quel vettore di valori un punto fisso della classe di pagine attualmente analizzata.

**5.1.4 Risultati restituiti dall’analisi dei punti fissi**

Dall’analisi vengo restituiti le regole XPath candidate che si sono rivelate essere punti fissi della classe di pagine, suddivisi tra costanti e variabili a seconda dei valori estratti.

Nel caso della ricerca dei punti fissi navigazionali viene tutta restituito anche l’insieme di riferimenti alle pagine HTML a cui indirizzano i valori estratti dalle regole.

**5.2 Analisi dei Punti Fissi navigazionali**

Il primo passo prevede un’analisi dei punti fissi navigazionali, detti NFP (*Navigational Fixed Point*). Una volta effettuata la ricerca dei punti fissi navigazionali vengono isolati e conservati solamente i punti fissi, dopo una rimozione di eventuali collegamenti a pagine non facenti parte dell’insieme iniziale di pagine HTML analizzato, che permettono la navigazione verso una delle altre classi di pagine.

Identificato un collegamento verso una classe di pagine lo si ritiene valido anche per le classi di pagine di cui la classe trovata è raffinamento. Ciò perché è possibile che da una determinata classe di pagine sia accessibile solamente una parte di un’altra classe di pagine, quindi non tutte le pagine HTML che la compongono.

Identificare come classe di pagine a cui indirizza quel punto fisso solamente la classe di pagine parziale, di cui i riferimenti sono effettivamente contenuti nella classe di pagine corrente, provocherebbe il rischio di ottenere una soluzione “ottima” più fine del necessario, ovvero con ulteriori suddivisioni nelle classi di pagine, si può ad esempio pensare ad un negozio online che abbia differenti classi di pagine contenenti collegamenti ai prodotti ma una sola classe di pagine per i prodotti, analizzando una delle classi di pagine contenente i collegamenti si riuscirebbe ad identificare la classe di pagine dei prodotti solo parzialmente, portando alla errata conclusione che i prodotti contenuti nei collegamenti delle diverse classi di pagine appartengano a classi di pagine differenti.

*public void executeNFP(Set<PageClass> pageClasses, Set<Page> pages){*

*for(PageClass pageClass : pageClasses){*

*Map<Set<String>, int[]> NFP = NFP(pageClass,pages)*

*if (NFP!= null) {*

*for(Set<String> IDs : NFP.keySet()){*

*Set<PageClass> NavClasses = PageClass.getPageClassesFromIDs(pageClasses,pcID)*

*if(NavClasses != null)*

*if(!NavClasses.isEmpty()){*

*NavClasses.remove(pageClass)*

*pageClass.addNFP(NavClasses,NFP.get(IDs))*

*}*

*}*

*}*

*}*

*}*

Si procede quindi a valutare la presenza di eventuali punti fissi navigazionali interni alle partizioni. Per ogni partizione vengono cercati i punti fissi che, generati su una classe di pagine appartenente alla partizione, conducono ad un’altra classe di pagine anch’essa appartenente alla partizione. I punti fissi navigazionali cosi trovati, detti punti fissi navigazionali compatibili, sono un efficace metodo per effettuare una prima selezione delle partizioni candidate ad essere “l’ottimo”. Si isolano dunque le partizioni che presentino il maggior numero medio di NFP tra le classi di pagine che lo compongono, tralasciando i punti fissi.

*for (Partition p : lattice.getPartitions()){*

*int[] total;*

*for(PageClass pageClass : p.getPageClasses){*

*for(Set<PageClass> pointsTo : pageClass.getNFP().keySet()){*

*for(PageClass pageClass2 : pointsTo){*

*if(p.getPageClasses().contains(pageClass2)){*

*total += pageClass.getNFP().get(pointsTo)*

*}*

*}*

*}*

*}*

*}*

**5.3 Analisi dei Punti Fissi di dati**

Si procede quindi alla ricerca di punti fissi di dati, DFP (*Data fixed point*) sulle partizioni isolate che presentano il più alto numero di punti fissi navigazionali. Vengono ricercati i punti fissi per ognuna delle classi di pagini che compongono le partizioni con il maggior numero di punti fissi navigazionali. Vengono selezionati solamente i punti fissi che generano valori per ogni pagina della classe analizzata, si trascurano quindi i punti fissi opzionali. Si rimuovono poi i punti fissi di sito, ovvero i punti fissi presenti ogni pagina del dataset iniziale.

Vengono poi selezionate le partizioni con il più alto numero di punti fissi navigazioni al fine di ridurre le partizioni sottoposte alla ricerca di punti fissi di dati. Tale scrematura consente una sostanziale riduzione in termini di tempi di esecuzione ed essendo la navigabilità un importante indice di valutazione dell’ottimo rimuovere le partizioni con una scarsa navigabilità interna non influisce sul risultato finale.

*public void executeDFP(Set<PageClass> pageClasses, Set<Page> pages, Set<FixedPoint<String>> siteDFP){*

*Map<FixedPoint<String>, PageClass> FP2PC*

*for(PageClass pc : pageClasses) {*

*Set<FixedPoint<String>> FixedPoints = DFP(XFParguments,pc,pages,range);*

*for(FixedPoint<String> fp : FixedPoints) {*

*if(!siteDFP.contains(fp)) {*

*FP2PC = addFixedPoints(FP2PC, fp, pc);*

*}*

*}*

*}*

*addToPageClass(FP2PC);*

*}*

*}*

Si assegna dunque ad ognuna delle partizioni un punteggio dato dalla media di punti fissi trovati nelle classi di pagine che la compongono, tenendo conto di una differenza di rilevanza tra punti fissi costanti e variabili ed escludendo nuovamente eventuali punti fissi “opzionali”. La partizione ottima individuata dall’algoritmo, e quindi la suddivisione delle pagine HTML che l’algoritmo restituisce, corrisponde con la partizione dal punteggio più alto.

**5.4 Ranking delle partizioni**

Si calcola quindi per ognuna delle partizioni rimanenti al calcolo di un *rank* per assegnare una valutazione a punti fissi navigazionali e di dato. Vengono esclusi dal calcolo per il *rank* i punti fissi opzionali, non indicativi nell’ambito dell’intera classe di pagine a cui appartengono, e viene considerato, nell’ambito dei punti fissi di dati, un peso differente a punti fissi variabili e costanti.

L’algoritmo sviluppato è configurabile in termini di numero di migliori risultati restituibili. Si può quindi configurare l’algoritmo per restituire tutte le partizioni aventi gli *n* migliori *rank*. Ciò perché il risultato “ottimo”, ovvero il partizionamento del set di pagine iniziale ricercato, può dipendere da specifiche esigenze dell’utilizzatore, non esiste quindi un unico partizionamento “perfetto”, e lo specifico partizionamento richiesto, dipendente magari da particolari esigenze, potrebbe non essere necessariamente quello con il maggior *rank*. Si lascia in questo modo all’utente finale di scegliere tra più alternative, più o meno valide.

Come si vedrà nel successivo capitolo, riguardante gli esperimenti effettuati, tale soluzione consente di aumentare precisione e recall dell’algoritmo.

# Capitolo 6 Esperimenti

Sono qui riportati i dati e le considerazioni inerenti agli esperimenti eseguiti sull’algoritmo.

Gli esperimenti sono stati eseguiti a due livelli di “sensibilità” del frammento XPath, utilizzando la stessa sensibilità sia per l’estrazione di regole dalle pagine sia per la generazione di regole a partire da vettori di valori.

Gli esperimenti sono stati eseguiti configurando l’algoritmo a restituire le partizioni con il migliore*¸* i migliori due ed i migliori tre *rank.* Per ognuna delle esecuzioni è stata individuata ed isolata la migliore delle partizioni proposte.

**6.1 Dataset**

Gli esperimenti sono su un insieme di 50 siti reali di cui sono state analizzate in media 45 pagine. Per ognuno dei siti web è stato realizzato un *golden* ovvero un ottimo ideale, partizionando le pagine secondo l’output desiderato. Dopo l’esecuzione dell’algoritmo su ognuno dei siti l’output è stato confrontato con il golden.

Viene qui di seguito riportato l’elenco completo di siti web analizzati negli esperimenti con il relativo numero di pagine e il numero di classi di pagine presenti nel *golden* di quel sito, ovvero in quanti classi di pagine si desidera partizionare l’insieme di pagine.

Tabella 1- Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Sito** | **homePage** | **#Pag** | **#Classi** |
| 1 | altaRimini | https://www.altarimini.it | 52 | 3 |
| 2 | altoAdige | https://www.altoadige.it | 49 | 3 |
| 3 | aranzulla | https://www.aranzulla.it | 45 | 3 |
| 4 | basilicataNotizie | http://www.basilicatanotizie.net | 51 | 3 |
| 5 | bofrost | https://www.bofrost.it/ | 45 | 4 |
| 6 | campaniaNotizie | http://www.campanianotizie.com | 51 | 3 |
| 7 | casertaWeb | https://casertaweb.com | 55 | 3 |
| 8 | castelliNotizie | https://www.castellinotizie.it/ | 49 | 3 |
| 9 | cinecittaShop | https://www.cinecittashop.it | 45 | 3 |
| 10 | cortilia | https://www.cortilia.it | 40 | 3 |
| 11 | direttaSicilia | https://www.direttasicilia.it | 45 | 3 |
| 12 | eataly | https://today.eataly.net/roma\_it | 38 | 3 |
| 13 | ecoDiTrapani | https://www.ecoditrapani.it/ | 65 | 3 |
| 14 | ecoFarma | https://www.ecofarma.it | 45 | 4 |
| 15 | ekathimerini | http://www.ekathimerini.com | 50 | 3 |
| 16 | eliteSurgelati | http://www.elitesurgelati.it | 45 | 3 |
| 17 | findus | https://www.findus.it | 45 | 4 |
| 18 | gDS | https://gds.it | 55 | 6 |
| 19 | giornaleDiSondrio | https://giornaledisondrio.it | 53 | 3 |
| 20 | ilFriuli | https://www.ilfriuli.it | 45 | 3 |
| 21 | ilGiornaleDiRieti | https://www.ilgiornaledirieti.it/index.asp | 39 | 3 |
| 22 | ilMattinoDiFoggia | https://www.ilmattinodifoggia.it/ | 35 | 3 |
| 23 | jb | https://www.jb.com.br | 50 | 3 |
| 24 | krone.at | https://www.krone.at | 45 | 3 |
| 25 | laSpesaCasa | https://www.laspesacasa.it | 40 | 3 |
| 26 | leCronacheLucane | http://www.lecronachelucane.it | 47 | 3 |
| 27 | libraccio | https://www.libraccio.it | 40 | 3 |
| 28 | libreriaUniversitaria | https://www.libreriauniversitaria.it | 45 | 3 |
| 29 | liguriaNotizie | https://www.ligurianotizie.it | 60 | 3 |
| 30 | marcheNotizie | https://www.marchenotizie.it | 51 | 3 |
| 31 | mobileShop | https://it.mobileshop.eu | 40 | 3 |
| 32 | notizieMolise | https://www.notiziemolise.it | 48 | 3 |
| 33 | paniereBio | https://www.panierebio.com | 45 | 3 |
| 34 | PiumeShopOnlie | https://www.piumeshoponline.com | 45 | 3 |
| 35 | playstore | https://play.google.com/store/apps | 45 | 2 |
| 36 | pskMegaStore | https://pskmegastore.com | 40 | 4 |
| 37 | quotidianoDiGela | https://www.quotidianodigela.it/ | 50 | 3 |
| 38 | quotidianoPiemontese | https://www.quotidianopiemontese.it | 48 | 3 |
| 39 | recensioniLibri | https://www.recensionilibri.org | 39 | 3 |
| 40 | spesaOnline24 | https://www.spesaonline24.it | 45 | 3 |
| 41 | theMovieDB | https://www.themoviedb.org | 43 | 6 |
| 42 | timesOfIndia | https://recipes.timesofindia.com | 45 | 3 |
| 43 | turismo | https://www.turismo.it | 35 | 3 |
| 44 | unilibro | https://www.unilibro.it | 42 | 3 |
| 45 | unioneSarda | https://www.unionesarda.it | 45 | 6 |
| 46 | utensileriaOnline | https://www.utensileriaonline.it | 33 | 4 |
| 47 | valdelsa | https://www.valdelsa.net/ | 47 | 3 |
| 48 | vivereMarche | https://www.viveremarche.it | 45 | 4 |
| 49 | wmePoint | https://www.wmepoint.com | 40 | 3 |
| 50 | workshop | https://www.workshopitaly.net | 33 | 4 |
| AVG | |  | 45,46 | 3,30 |

**6.2 Confronto con il *golden***

Il confronto tra l’output e il golden viene effettuato per ogni singola classe di pagine, calcolando poi una media.

Si procede prendendo, a turno, per ognuna delle classi del *golden* la classe di pagine dell’output che genera il maggior valori di F1 score. Una volta assegnata una classe dell’output non è più possibile assegnare quella stessa classe ad un'altra classe del *golden.*

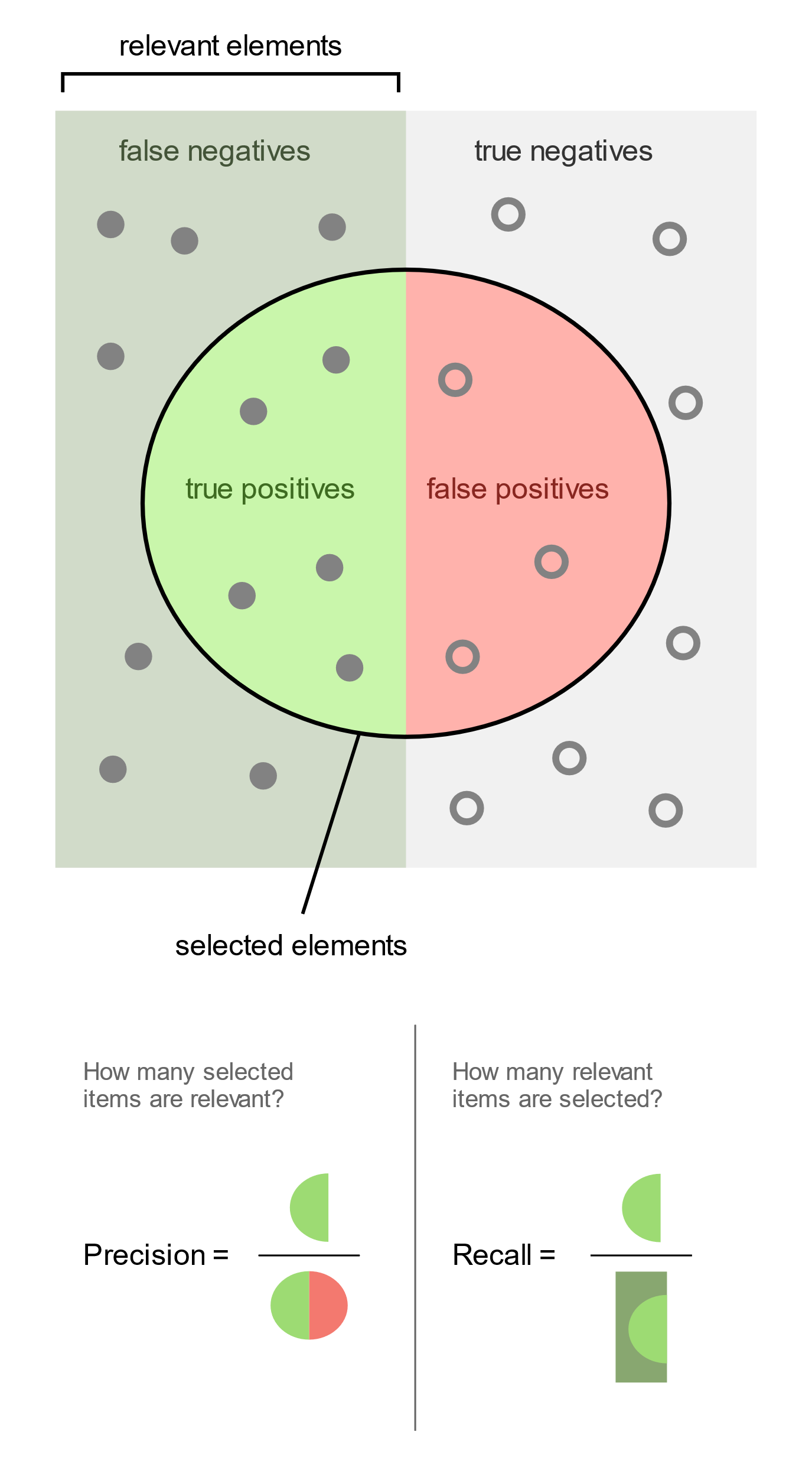
****6.3 Precision, recall e F1 score**

Figura 9- Precision e recall

La misura del confronto viene effettuata in termini di *precision* e *recall,* trattate nei successivi paragrafi e di cui la Figura 9[[6]](#footnote-6) risulta chiarificatrice, da cui è poi possibile calcolare un indice unico, tenente conto di entrambi, chiamato *F1 score.*

**6.3.1 Precision**

La *precision*, o precisione, è appunto un indice di precisione. Si misura quindi quanti degli elementi presenti nella classe di pagine dell’output sia presenti nella classe, si ottiene quindi dividendo il numero di risultati veri positivi, ovvero il numero di pagine presenti nella classe di pagine dell’output presenti anche in quella del *golden*, per il numero totale di pagine presenti nella classe dell’output, composta quindi da veri positivi più falsi positivi, elementi presenti nell’output ma non nel *golden*.

**6.3.2 Recall**

La *recall*, o richiamo, è invece una misura di completezza, ovvero quanti degli elementi che avrei dovuto trovare ho realmente trovato. Si ottiene quindi dividendo il numero di veri positivi per il numero di veri positivi più il numero di falsi negativi, ovvero il numero di elementi presenti nella classe del *golden* che non sono però presenti in quella dell’output.

**6.3.3 F1 score**

La F1 score è una media armonica tenente conto di entrambe precisione e recall. È in generale possibile calcolare un F-Score dando un peso differente a *precision* e *recall* secondo la formula:

Da cui assegnando a β un valore maggiore di uno si assegna un peso maggiore alla precisione e viceversa assegnandogli un valore minore di uno si considera un peso maggiore per il richiamo.

Tuttavia, nell’algoritmo è stato assegnato a β il valore di 1 per dare lo stesso peso a precisione e recall, ottenendo:

**6.4 Risultati degli esperimenti**

Verranno qui di seguito riportati i risultati ottenuti dall’esecuzione dell’algoritmo sul dataset precedentemente descritto.

**6.4.1 Espressività 4**

Verranno qui riportati i risultati ottenuti con un range, ovvero una espressività del frammento XPath pari a 4.

Migliore *Rank:*

Tabella 2. Espressività 4 - Migliore Rank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Sito** | **Tempo(s)** | **Precision** | **Recall** | **F1** |
| 1 | altaRimini | 194 | 1,000 | 0,959 | 0,979 |
| 2 | altoAdige | 209 | 1,000 | 0,715 | 0,834 |
| 3 | aranzulla | 224 | 1,000 | 0,706 | 0,828 |
| 4 | basilicataNotizie | 393 | 0,357 | 0,342 | 0,350 |
| 5 | bofrost | 230 | 1,000 | 0,604 | 0,753 |
| 6 | campaniaNotizie | 279 | 0,342 | 0,348 | 0,345 |
| 7 | casertaWeb | 424 | 1,000 | 0,630 | 0,773 |
| 8 | castelliNotizie | 398 | 0,535 | 0,667 | 0,593 |
| 9 | cinecittaShop | 190 | 1,000 | 0,561 | 0,718 |
| 10 | cortilia | 176 | 1,000 | 0,535 | 0,697 |
| 11 | direttaSicilia | 218 | 0,652 | 0,667 | 0,659 |
| 12 | eataly | 319 | 1,000 | 0,714 | 0,833 |
| 13 | ecoDiTrapani | 336 | 0,964 | 0,923 | 0,943 |
| 14 | ecoFarma | 721 | 0,509 | 0,667 | 0,577 |
| 15 | ekathimerini | 350 | 0,833 | 0,938 | 0,882 |
| 16 | eliteSurgelati | 197 | 0,576 | 0,667 | 0,618 |
| 17 | findus | 402 | 1,000 | 0,526 | 0,689 |
| 18 | gDS | 369 | 0,676 | 0,750 | 0,711 |
| 19 | giornaleDiSondrio | 269 | 1,000 | 0,674 | 0,805 |
| 20 | ilFriuli | 255 | 1,000 | 1,000 | 1,000 |
| 21 | ilGiornaleDiRieti | 156 | 0,766 | 0,679 | 0,720 |
| 22 | ilMattinoDiFoggia | 465 | 1,000 | 0,889 | 0,941 |
| 23 | jb | 241 | 0,479 | 0,667 | 0,558 |
| 24 | krone.at | 1796 | 1,000 | 0,969 | 0,984 |
| 25 | laSpesaCasa | 197 | 0,650 | 0,667 | 0,658 |
| 26 | leCronacheLucane | 335 | 1,000 | 0,833 | 0,909 |
| 27 | libraccio | 343 | 1,000 | 1,000 | 1,000 |
| 28 | libreriaUniversitaria | 447 | 1,000 | 1,000 | 1,000 |
| 29 | liguriaNotizie | 698 | 1,000 | 0,410 | 0,581 |
| 30 | marcheNotizie | 608 | 1,000 | 0,492 | 0,660 |
| 31 | mobileShop | 264 | 0,573 | 0,667 | 0,616 |
| 32 | notizieMolise | 399 | 1,000 | 0,717 | 0,835 |
| 33 | paniereBio | 385 | 0,833 | 0,622 | 0,712 |
| 34 | PiumeShopOnlie | 638 | 1,000 | 0,939 | 0,969 |
| 35 | playstore | 305 | 1,000 | 0,595 | 0,746 |
| 36 | pskMegaStore | 1408 | 0,761 | 0,758 | 0,760 |
| 37 | quotidianoDiGela | 513 | 1,000 | 0,681 | 0,810 |
| 38 | quotidianoPiemontese | 175 | 0,355 | 0,667 | 0,463 |
| 39 | recensioniLibri | 285 | 0,518 | 0,667 | 0,583 |
| 40 | spesaOnline24 | 208 | 0,644 | 0,667 | 0,655 |
| 41 | theMovieDB | 628 | 0,505 | 0,354 | 0,416 |
| 42 | timesOfIndia | 496 | 1,000 | 0,899 | 0,947 |
| 43 | turismo | 259 | 1,000 | 0,708 | 0,829 |
| 44 | unilibro | 453 | 1,000 | 0,681 | 0,810 |
| 45 | unioneSarda | 534 | 0,990 | 0,690 | 0,813 |
| 46 | utensileriaOnline | 188 | 1,000 | 0,725 | 0,841 |
| 47 | valdelsa | 258 | 1,000 | 0,889 | 0,941 |
| 48 | vivereMarche | 1056 | 1,000 | 0,781 | 0,877 |
| 49 | wmePoint | 1266 | 1,000 | 0,712 | 0,832 |
| 50 | workshop | 3733 | 0,259 | 0,259 | 0,259 |
| AVG | | 497,80 | 0,835 | 0,697 | 0,746 |
|  | Total time -> | 24890 | F1 on AVG-> | | 0,760 |

Migliori 2 *Rank*:

Tabella 3 - Espressività 4 - Migliori 2 Rank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Sito** | **Tempo(s)** | **Precision** | **Recall** | **F1** |
| 1 | altaRimini | 220 | 1,000 | 0,959 | 0,979 |
| 2 | altoAdige | 283 | 1,000 | 0,715 | 0,834 |
| 3 | aranzulla | 322 | 1,000 | 0,706 | 0,828 |
| 4 | basilicataNotizie | 397 | 0,357 | 0,477 | 0,409 |
| 5 | bofrost | 277 | 1,000 | 0,604 | 0,753 |
| 6 | campaniaNotizie | 286 | 1,000 | 0,913 | 0,955 |
| 7 | casertaWeb | 543 | 1,000 | 0,630 | 0,773 |
| 8 | castelliNotizie | 436 | 1,000 | 0,971 | 0,985 |
| 9 | cinecittaShop | 209 | 1,000 | 0,561 | 0,718 |
| 10 | cortilia | 176 | 1,000 | 0,535 | 0,697 |
| 11 | direttaSicilia | 231 | 1,000 | 0,833 | 0,909 |
| 12 | eataly | 345 | 1,000 | 0,714 | 0,833 |
| 13 | ecoDiTrapani | 426 | 0,987 | 0,910 | 0,947 |
| 14 | ecoFarma | 872 | 0,509 | 0,667 | 0,577 |
| 15 | ekathimerini | 367 | 0,833 | 0,938 | 0,882 |
| 16 | eliteSurgelati | 207 | 0,990 | 0,944 | 0,967 |
| 17 | findus | 426 | 1,000 | 0,756 | 0,861 |
| 18 | gDS | 356 | 0,676 | 0,750 | 0,711 |
| 19 | giornaleDiSondrio | 295 | 1,000 | 0,674 | 0,805 |
| 20 | ilFriuli | 329 | 1,000 | 1,000 | 1,000 |
| 21 | ilGiornaleDiRieti | 164 | 1,000 | 1,000 | 1,000 |
| 22 | ilMattinoDiFoggia | 481 | 1,000 | 0,889 | 0,941 |
| 23 | jb | 251 | 1,000 | 0,759 | 0,863 |
| 24 | krone.at | 2172 | 1,000 | 0,969 | 0,984 |
| 25 | laSpesaCasa | 224 | 1,000 | 0,833 | 0,909 |
| 26 | leCronacheLucane | 403 | 1,000 | 0,833 | 0,909 |
| 27 | libraccio | 532 | 1,000 | 1,000 | 1,000 |
| 28 | libreriaUniversitaria | 581 | 1,000 | 1,000 | 1,000 |
| 29 | liguriaNotizie | 759 | 1,000 | 0,547 | 0,707 |
| 30 | marcheNotizie | 727 | 1,000 | 0,590 | 0,742 |
| 31 | mobileShop | 419 | 0,988 | 0,946 | 0,966 |
| 32 | notizieMolise | 574 | 1,000 | 0,800 | 0,889 |
| 33 | paniereBio | 492 | 0,833 | 0,733 | 0,780 |
| 34 | PiumeShopOnlie | 1007 | 1,000 | 0,990 | 0,995 |
| 35 | playstore | 383 | 1,000 | 0,595 | 0,746 |
| 36 | pskMegaStore | 1817 | 0,761 | 0,758 | 0,760 |
| 37 | quotidianoDiGela | 636 | 1,000 | 0,681 | 0,810 |
| 38 | quotidianoPiemontese | 180 | 1,000 | 1,000 | 1,000 |
| 39 | recensioniLibri | 435 | 1,000 | 0,607 | 0,755 |
| 40 | spesaOnline24 | 294 | 1,000 | 0,683 | 0,812 |
| 41 | theMovieDB | 640 | 0,505 | 0,424 | 0,461 |
| 42 | timesOfIndia | 718 | 1,000 | 0,899 | 0,947 |
| 43 | turismo | 382 | 1,000 | 0,808 | 0,894 |
| 44 | unilibro | 580 | 1,000 | 0,681 | 0,810 |
| 45 | unioneSarda | 559 | 0,977 | 0,722 | 0,830 |
| 46 | utensileriaOnline | 219 | 1,000 | 0,881 | 0,937 |
| 47 | valdelsa | 277 | 1,000 | 1,000 | 1,000 |
| 48 | vivereMarche | 1058 | 1,000 | 0,781 | 0,877 |
| 49 | wmePoint | 1290 | 1,000 | 0,833 | 0,909 |
| 50 | workshop | 5318 | 0,731 | 0,716 | 0,723 |
| AVG | | 611,50 | 0,943 | 0,784 | 0,848 |
|  | Total time -> | 30575 | F1 on AVG-> | | 0,856 |

Migliori 3 *Rank:*

Tabella 4 - Espressività 4 - Migliori 3 Rank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Sito** | **Tempo(s)** | **Precision** | **Recall** | **F1** |
| 1 | altaRimini | 222 | 1,000 | 0,959 | 0,979 |
| 2 | altoAdige | 280 | 1,000 | 0,715 | 0,834 |
| 3 | aranzulla | 325 | 1,000 | 0,706 | 0,828 |
| 4 | basilicataNotizie | 385 | 0,357 | 0,477 | 0,409 |
| 5 | bofrost | 277 | 1,000 | 0,604 | 0,753 |
| 6 | campaniaNotizie | 350 | 1,000 | 0,913 | 0,955 |
| 7 | casertaWeb | 507 | 1,000 | 0,630 | 0,773 |
| 8 | castelliNotizie | 484 | 1,000 | 0,971 | 0,985 |
| 9 | cinecittaShop | 207 | 1,000 | 0,561 | 0,718 |
| 10 | cortilia | 180 | 1,000 | 0,535 | 0,697 |
| 11 | direttaSicilia | 310 | 1,000 | 0,833 | 0,909 |
| 12 | eataly | 343 | 1,000 | 0,714 | 0,833 |
| 13 | ecoDiTrapani | 431 | 0,987 | 0,910 | 0,947 |
| 14 | ecoFarma | 922 | 0,509 | 0,717 | 0,595 |
| 15 | ekathimerini | 355 | 0,833 | 0,938 | 0,882 |
| 16 | eliteSurgelati | 318 | 0,990 | 0,972 | 0,981 |
| 17 | findus | 564 | 1,000 | 0,756 | 0,861 |
| 18 | gDS | 474 | 0,676 | 0,750 | 0,711 |
| 19 | giornaleDiSondrio | 328 | 1,000 | 0,674 | 0,805 |
| 20 | ilFriuli | 388 | 1,000 | 1,000 | 1,000 |
| 21 | ilGiornaleDiRieti | 192 | 1,000 | 1,000 | 1,000 |
| 22 | ilMattinoDiFoggia | 592 | 1,000 | 0,889 | 0,941 |
| 23 | jb | 381 | 1,000 | 0,759 | 0,863 |
| 24 | krone.at | 2281 | 1,000 | 0,969 | 0,984 |
| 25 | laSpesaCasa | 229 | 1,000 | 0,833 | 0,909 |
| 26 | leCronacheLucane | 358 | 1,000 | 0,833 | 0,909 |
| 27 | libraccio | 487 | 1,000 | 1,000 | 1,000 |
| 28 | libreriaUniversitaria | 568 | 1,000 | 1,000 | 1,000 |
| 29 | liguriaNotizie | 741 | 1,000 | 0,547 | 0,707 |
| 30 | marcheNotizie | 700 | 1,000 | 0,590 | 0,742 |
| 31 | mobileShop | 300 | 0,988 | 0,958 | 0,973 |
| 32 | notizieMolise | 454 | 1,000 | 0,800 | 0,889 |
| 33 | paniereBio | 416 | 0,833 | 0,733 | 0,780 |
| 34 | PiumeShopOnlie | 724 | 1,000 | 1,000 | 1,000 |
| 35 | playstore | 377 | 1,000 | 0,595 | 0,746 |
| 36 | pskMegaStore | 1726 | 0,761 | 0,758 | 0,760 |
| 37 | quotidianoDiGela | 604 | 1,000 | 0,681 | 0,810 |
| 38 | quotidianoPiemontese | 242 | 1,000 | 1,000 | 1,000 |
| 39 | recensioniLibri | 427 | 1,000 | 0,607 | 0,755 |
| 40 | spesaOnline24 | 297 | 1,000 | 0,756 | 0,861 |
| 41 | theMovieDB | 675 | 0,505 | 0,424 | 0,461 |
| 42 | timesOfIndia | 660 | 1,000 | 0,899 | 0,947 |
| 43 | turismo | 335 | 1,000 | 1,000 | 1,000 |
| 44 | unilibro | 560 | 1,000 | 0,681 | 0,810 |
| 45 | unioneSarda | 564 | 0,977 | 0,722 | 0,830 |
| 46 | utensileriaOnline | 218 | 1,000 | 0,881 | 0,937 |
| 47 | valdelsa | 261 | 1,000 | 1,000 | 1,000 |
| 48 | vivereMarche | 1165 | 1,000 | 0,781 | 0,877 |
| 49 | wmePoint | 1220 | 1,000 | 0,833 | 0,909 |
| 50 | workshop | 5416 | 1,000 | 0,866 | 0,928 |
| AVG | | 616,40 | 0,948 | 0,795 | 0,856 |
|  | Total time -> | 30820 | F1 on AVG-> | | 0,865 |

**6.4.2 Espressività 6**

Verranno qui riportati i risultati ottenuti con un range, ovvero una espressività del frammento XPath pari a 6.

Migliore *Rank:*

Tabella 5 - Espressività 6 - Migliore Rank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Sito** | **Tempo(s)** | **Precision** | **Recall** | **F1** |
| 1 | altaRimini | 220 | 0,310 | 0,500 | 0,383 |
| 2 | altoAdige | 254 | 1,000 | 0,741 | 0,851 |
| 3 | aranzulla | 722 | 1,000 | 0,706 | 0,828 |
| 4 | basilicataNotizie | 434 | 1,000 | 0,923 | 0,960 |
| 5 | bofrost | 222 | 1,000 | 0,604 | 0,753 |
| 6 | campaniaNotizie | 297 | 0,341 | 0,348 | 0,344 |
| 7 | casertaWeb | 496 | 1,000 | 0,679 | 0,809 |
| 8 | castelliNotizie | 476 | 0,535 | 0,667 | 0,593 |
| 9 | cinecittaShop | 194 | 1,000 | 0,561 | 0,718 |
| 10 | cortilia | 212 | 1,000 | 0,579 | 0,733 |
| 11 | direttaSicilia | 458 | 0,652 | 0,667 | 0,659 |
| 12 | eataly | 356 | 1,000 | 0,681 | 0,810 |
| 13 | ecoDiTrapani | 411 | 0,006 | 0,333 | 0,011 |
| 14 | ecoFarma | 8264 | 0,509 | 0,550 | 0,529 |
| 15 | ekathimerini | 389 | 1,000 | 0,903 | 0,949 |
| 16 | eliteSurgelati | 200 | 0,576 | 0,667 | 0,618 |
| 17 | findus | 437 | 1,000 | 0,526 | 0,689 |
| 18 | gDS | 11442 | 0,676 | 0,833 | 0,746 |
| 19 | giornaleDiSondrio | 384 | 1,000 | 0,674 | 0,805 |
| 20 | ilFriuli | 267 | 1,000 | 1,000 | 1,000 |
| 21 | ilGiornaleDiRieti | 176 | 0,766 | 0,679 | 0,720 |
| 22 | ilMattinoDiFoggia | 569 | 1,000 | 0,889 | 0,941 |
| 23 | jb | 1123 | 0,479 | 0,667 | 0,558 |
| 24 | krone.at | 1581 | 1,000 | 0,969 | 0,984 |
| 25 | laSpesaCasa | 265 | 0,650 | 0,667 | 0,658 |
| 26 | leCronacheLucane | 566 | 1,000 | 0,833 | 0,909 |
| 27 | libraccio | 586 | 1,000 | 0,711 | 0,831 |
| 28 | libreriaUniversitaria | 587 | 1,000 | 1,000 | 1,000 |
| 29 | liguriaNotizie | 9595 | 1,000 | 0,482 | 0,651 |
| 30 | marcheNotizie | 717 | 1,000 | 0,492 | 0,660 |
| 31 | mobileShop | 320 | 0,573 | 0,667 | 0,616 |
| 32 | notizieMolise | 558 | 1,000 | 0,717 | 0,835 |
| 33 | paniereBio | 368 | 0,342 | 0,400 | 0,369 |
| 34 | PiumeShopOnlie | 597 | 1,000 | 0,939 | 0,969 |
| 35 | playstore | 295 | 1,000 | 0,595 | 0,746 |
| 36 | pskMegaStore | 1497 | 0,761 | 0,633 | 0,691 |
| 37 | quotidianoDiGela | 567 | 1,000 | 0,681 | 0,810 |
| 38 | quotidianoPiemontese | 255 | 0,355 | 0,667 | 0,463 |
| 39 | recensioniLibri | 5442 | 0,518 | 0,667 | 0,583 |
| 40 | spesaOnline24 | 252 | 0,644 | 0,667 | 0,655 |
| 41 | theMovieDB | 878 | 0,380 | 0,313 | 0,343 |
| 42 | timesOfIndia | 775 | 1,000 | 0,899 | 0,947 |
| 43 | turismo | 564 | 0,412 | 0,667 | 0,509 |
| 44 | unilibro | 2033 | 1,000 | 0,681 | 0,810 |
| 45 | unioneSarda | 755 | 0,944 | 0,773 | 0,850 |
| 46 | utensileriaOnline | 229 | 1,000 | 0,881 | 0,937 |
| 47 | valdelsa | 2276 | 0,355 | 0,667 | 0,463 |
| 48 | vivereMarche | 1631 | 1,000 | 0,802 | 0,890 |
| 49 | wmePoint | 1557 | 1,000 | 0,712 | 0,832 |
| 50 | workshop | 4429 | 0,259 | 0,259 | 0,259 |
| AVG | | 1343,56 | 0,781 | 0,676 | 0,706 |
|  | Total Time-> | 67178 | F1 on AVG-> | | 0,725 |

Migliori 2 *Rank:*

Tabella 6 - Espressività 6 - Migliori 2 Rank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Sito** | **Tempo(s)** | **Precision** | **Recall** | **F1** |
| 1 | altaRimini | 232 | 1,000 | 0,979 | 0,989 |
| 2 | altoAdige | 256 | 1,000 | 0,741 | 0,851 |
| 3 | aranzulla | 3602 | 1,000 | 0,706 | 0,828 |
| 4 | basilicataNotizie | 507 | 1,000 | 1,000 | 1,000 |
| 5 | bofrost | 282 | 1,000 | 0,604 | 0,753 |
| 6 | campaniaNotizie | 299 | 1,000 | 0,913 | 0,955 |
| 7 | casertaWeb | 687 | 1,000 | 0,679 | 0,809 |
| 8 | castelliNotizie | 528 | 1,000 | 0,971 | 0,985 |
| 9 | cinecittaShop | 210 | 1,000 | 0,561 | 0,718 |
| 10 | cortilia | 205 | 1,000 | 0,579 | 0,733 |
| 11 | direttaSicilia | 876 | 1,000 | 0,690 | 0,817 |
| 12 | eataly | 347 | 1,000 | 0,681 | 0,810 |
| 13 | ecoDiTrapani | 473 | 0,964 | 0,923 | 0,943 |
| 14 | ecoFarma | 8254 | 0,509 | 0,667 | 0,577 |
| 15 | ekathimerini | 452 | 1,000 | 0,903 | 0,949 |
| 16 | eliteSurgelati | 215 | 0,990 | 0,917 | 0,952 |
| 17 | findus | 492 | 1,000 | 0,810 | 0,895 |
| 18 | gDS | 11348 | 0,676 | 0,833 | 0,746 |
| 19 | giornaleDiSondrio | 492 | 1,000 | 0,924 | 0,960 |
| 20 | ilFriuli | 349 | 1,000 | 1,000 | 1,000 |
| 21 | ilGiornaleDiRieti | 170 | 1,000 | 1,000 | 1,000 |
| 22 | ilMattinoDiFoggia | 541 | 1,000 | 0,889 | 0,941 |
| 23 | jb | 1295 | 1,000 | 0,843 | 0,915 |
| 24 | krone.at | 2346 | 1,000 | 0,969 | 0,984 |
| 25 | laSpesaCasa | 294 | 1,000 | 0,833 | 0,909 |
| 26 | leCronacheLucane | 794 | 1,000 | 0,833 | 0,909 |
| 27 | libraccio | 571 | 1,000 | 0,711 | 0,831 |
| 28 | libreriaUniversitaria | 576 | 1,000 | 1,000 | 1,000 |
| 29 | liguriaNotizie | 9443 | 1,000 | 0,547 | 0,707 |
| 30 | marcheNotizie | 879 | 1,000 | 0,590 | 0,742 |
| 31 | mobileShop | 422 | 0,988 | 0,946 | 0,966 |
| 32 | notizieMolise | 546 | 1,000 | 0,767 | 0,868 |
| 33 | paniereBio | 457 | 0,833 | 0,733 | 0,780 |
| 34 | PiumeShopOnlie | 681 | 1,000 | 0,990 | 0,995 |
| 35 | playstore | 321 | 1,000 | 0,595 | 0,746 |
| 36 | pskMegaStore | 1840 | 0,761 | 0,758 | 0,760 |
| 37 | quotidianoDiGela | 626 | 1,000 | 0,681 | 0,810 |
| 38 | quotidianoPiemontese | 296 | 1,000 | 1,000 | 1,000 |
| 39 | recensioniLibri | 11022 | 0,944 | 0,574 | 0,714 |
| 40 | spesaOnline24 | 307 | 1,000 | 0,683 | 0,812 |
| 41 | theMovieDB | 705 | 0,380 | 0,313 | 0,343 |
| 42 | timesOfIndia | 699 | 1,000 | 0,899 | 0,947 |
| 43 | turismo | 431 | 1,000 | 1,000 | 1,000 |
| 44 | unilibro | 1962 | 1,000 | 0,681 | 0,810 |
| 45 | unioneSarda | 697 | 0,977 | 0,771 | 0,862 |
| 46 | utensileriaOnline | 222 | 1,000 | 0,881 | 0,937 |
| 47 | valdelsa | 1786 | 1,000 | 1,000 | 1,000 |
| 48 | vivereMarche | 1341 | 1,000 | 0,969 | 0,984 |
| 49 | wmePoint | 1294 | 1,000 | 0,833 | 0,909 |
| 50 | workshop | 4561 | 0,731 | 0,716 | 0,723 |
| AVG | | 1544,62 | 0,955 | 0,802 | 0,864 |
|  | Total Time-> | 77231 | F1 on AVG-> | | 0,872 |

Migliori 3 *Rank:*

Tabella 7 - Espressività 6 - Migliori 3 Rank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Sito** | **Tempo(s)** | **Precision** | **Recall** | **F1** |
| 1 | altaRimini | 227 | 1,000 | 0,979 | 0,989 |
| 2 | altoAdige | 252 | 1,000 | 0,741 | 0,851 |
| 3 | aranzulla | 3625 | 1,000 | 0,706 | 0,828 |
| 4 | basilicataNotizie | 483 | 1,000 | 1,000 | 1,000 |
| 5 | bofrost | 278 | 1,000 | 0,604 | 0,753 |
| 6 | campaniaNotizie | 347 | 1,000 | 0,913 | 0,955 |
| 7 | casertaWeb | 699 | 1,000 | 0,679 | 0,809 |
| 8 | castelliNotizie | 622 | 1,000 | 0,971 | 0,985 |
| 9 | cinecittaShop | 209 | 1,000 | 0,561 | 0,718 |
| 10 | cortilia | 209 | 1,000 | 0,579 | 0,733 |
| 11 | direttaSicilia | 884 | 1,000 | 1,000 | 1,000 |
| 12 | eataly | 424 | 1,000 | 0,681 | 0,810 |
| 13 | ecoDiTrapani | 482 | 0,964 | 0,923 | 0,943 |
| 14 | ecoFarma | 8604 | 0,509 | 0,667 | 0,577 |
| 15 | ekathimerini | 440 | 1,000 | 0,903 | 0,949 |
| 16 | eliteSurgelati | 217 | 0,990 | 0,944 | 0,967 |
| 17 | findus | 480 | 1,000 | 0,828 | 0,906 |
| 18 | gDS | 11172 | 0,676 | 0,833 | 0,746 |
| 19 | giornaleDiSondrio | 450 | 1,000 | 0,924 | 0,960 |
| 20 | ilFriuli | 336 | 1,000 | 1,000 | 1,000 |
| 21 | ilGiornaleDiRieti | 167 | 1,000 | 1,000 | 1,000 |
| 22 | ilMattinoDiFoggia | 717 | 1,000 | 0,889 | 0,941 |
| 23 | jb | 1276 | 1,000 | 0,843 | 0,915 |
| 24 | krone.at | 2237 | 1,000 | 0,969 | 0,984 |
| 25 | laSpesaCasa | 321 | 1,000 | 0,833 | 0,909 |
| 26 | leCronacheLucane | 789 | 1,000 | 0,833 | 0,909 |
| 27 | libraccio | 526 | 1,000 | 1,000 | 1,000 |
| 28 | libreriaUniversitaria | 880 | 1,000 | 1,000 | 1,000 |
| 29 | liguriaNotizie | 9711 | 1,000 | 0,547 | 0,707 |
| 30 | marcheNotizie | 926 | 1,000 | 0,590 | 0,742 |
| 31 | mobileShop | 438 | 0,988 | 0,946 | 0,966 |
| 32 | notizieMolise | 553 | 1,000 | 0,800 | 0,889 |
| 33 | paniereBio | 484 | 0,833 | 0,733 | 0,780 |
| 34 | PiumeShopOnlie | 817 | 1,000 | 1,000 | 1,000 |
| 35 | playstore | 318 | 1,000 | 0,595 | 0,746 |
| 36 | pskMegaStore | 1852 | 0,761 | 0,758 | 0,760 |
| 37 | quotidianoDiGela | 625 | 1,000 | 0,681 | 0,810 |
| 38 | quotidianoPiemontese | 289 | 1,000 | 1,000 | 1,000 |
| 39 | recensioniLibri | 11082 | 0,944 | 0,574 | 0,714 |
| 40 | spesaOnline24 | 300 | 1,000 | 0,756 | 0,861 |
| 41 | theMovieDB | 645 | 0,505 | 0,354 | 0,416 |
| 42 | timesOfIndia | 665 | 1,000 | 0,899 | 0,947 |
| 43 | turismo | 478 | 1,000 | 1,000 | 1,000 |
| 44 | unilibro | 1966 | 1,000 | 0,681 | 0,810 |
| 45 | unioneSarda | 728 | 0,977 | 0,771 | 0,862 |
| 46 | utensileriaOnline | 222 | 1,000 | 0,881 | 0,937 |
| 47 | valdelsa | 1742 | 1,000 | 1,000 | 1,000 |
| 48 | vivereMarche | 1359 | 1,000 | 0,969 | 0,984 |
| 49 | wmePoint | 1275 | 1,000 | 0,833 | 0,909 |
| 50 | workshop | 6996 | 1,000 | 0,866 | 0,928 |
| AVG | | 1616,48 | 0,963 | 0,821 | 0,878 |
|  | Total Time -> | 80824 | F1 on AVG-> | | 0,886 |

**6.5 Analisi dei risultati**

Verranno ora riportati alcuni grafici utili ad un confronto tra i dati ottenuti.

Figura 10- Grafico espressività 4

Figura 11- Grafico espressività 6

Figura 12- Grafico espressività 4 e 6

Figura 13- Grafico tempi di esecuzione

Si può quindi notare come l’incremento del numero di migliori *rank* da restituire, con il corrispettivo aumento delle partizioni restituite in output, porti ad un notevole incremento di prestazione in termini di F1 score. Si può inoltre notare come anche una maggiore espressività porti, a fronte di un notevole peggioramento in termini di tempi di esecuzione, ad un lieve incremento nella precisione e nel richiamo. Si ha un unico caso, che potrebbe risultare anomalo, in cui l’espressività posta ad un valore inferiore porta ad un risultato migliore. Tuttavia, è possibile notare che tale caso corrisponde alla restituzione di un unico valore di *rank* e si può quindi intuire che con una espressività minore il risultato restituito sia “meno specifico” e dunque un leggermente più ad alto livello nella valutazione delle differenze, restituendo quindi una partizione poche volte molto accurata ma raramente troppo distante da quella desiderata. Con un range più altro viceversa la partizione restituita sarà molto più “specifica” ottenendo in alcuni casi dei risultati discretamente elevati, se la direzione in cui l’output ha “approfondito” le differenze è la stessa utilizzata per stabilire il *golden*, ma che tuttavia risulta in un numero più elevato di casi meno precisa poiché la direzione di approfondimento è differente da quella desiderata, Si ha quindi che il valore medio di F1 score misurato risulta minore.

# Capitolo 7 Sviluppi futuri

Sono possibili molteplici sviluppi futuri sull’algoritmo sviluppato. È sicuramente modificare la generazione delle partizioni, riducendone il numero, introducendo un criterio per stabilire in fase di generazione quale tipo di partizione sia idonea rimuovendo preventivamente le partizioni meno adatte. Questa ottimizzazione potrebbe consentire l’esecuzione dell’algoritmo su porzioni maggiori di siti e, con alta probabilità, su siti interi senza incorrere in tempi di esecuzione eccessivamente elevati.

Si potrebbe inoltre lavorare sulla ricerca di un miglior compromesso tra l’espressività scelta per il frammento XPath e i tempi di esecuzione, che ne sono fortemente influenzati, anche in funzione del livello di dettaglio che l’utilizzatore desidera ottenere.

**Appendice**

**XPath**

**Struttura ad albero di file XML**

Preso un qualsiasi file XML è possibile rappresentarlo con una struttura ad albero in cui ogni elemento del file XML, ovvero ogni “blocco” racchiuso tra un tag di apertura ed uno di chiusura, è rappresentato da un nodo. Tutti gli elementi contenuti all’interno di quell’elemento saranno nodi figli di tale nodo. Ad ogni nodo vengono inoltre associati i suoi eventuali attributi. È possibile utilizzare una rappresentazione di questo tipo poiché i documenti XML rispettano una struttura gerarchica, e non è quindi possibile chiudere un elemento senza aver precedentemente chiuso tutti gli elementi aperti al suo interno. Deve quindi essere necessariamente presente un elemento che racchiude tutti gli altri elementi del file, che corrisponderà quindi ad un nodo, genitore di tutti gli altri nodi presenti nell’albero. Tale nodo prende il nome di nodo radice.

I nodi posso essere di vario tipo, il W3C (World Wide Web Consortium), l’organizzazione sviluppatrice degli XPath, ne identifica 7 tipologie:

-*Root node* , il nodo radice dell’intero file XML

-*Element node* , un nodo che rappresenta un elemento del file XML. Un elemento in XML è un blocco costruttivo, comprende tutto ciò che è racchiuso tra il tag di apertura e il tag di chiusura dell’elemento.

-*Text node,* un nodo di tipo testo rappresenta il testo contenuto in un elemento XML

-*Attribute node* , rappresenta un attributo di un elemento XML

-*Namespace node* , ogni nodo elemento ha un nodo di tipo *namespace* per ognuno dei prefissi usati per quell’elemento.

-*Processing instruction nodes* , rappresenta un’istruzione di elaborazione inserita nel file XML.

-*Comment node* , un nodo contenente un commento al fine XML.

**Struttura delle regole XPath**

Il linguaggio XPath è un linguaggio XML composto da espressioni. Presa la rappresentazione in forma di albero di un qualsiasi file XML un’espressione in XPath, detta anche regola, rappresenta un cammino a partire da un nodo iniziale o dalla radice del file fino a giungere ad uno o più nodi.

Una regola che inizia a partire da un nodo identifica un percorso relativo, mentre una regola che inizia a partire dalla radice del file identifica un percorso assoluto ed inizia con un carattere ‘/’.

Ogni regola è composta da più passaggi, o passi, ognuno di questi passaggi ha una struttura caratteristica analizzata nel dettaglio successivamente. Ognuno di questi passaggi corrisponde ad una serie di condizioni che un nodo deve rispettare per essere ritenuto valido. Si parte dunque dal nodo iniziale, o dalla radice del file, e si verificano i nodi che rispettano le condizioni del primo passaggio. Per ognuno dei nodi trovati viene creato un nuovo cammino, composto inizialmente solamente dal nodo iniziale e dal nodo trovato.

Si prosegue quindi ricorsivamente e si applicano le condizioni del passaggio successivo della regola all’ultimo nodo di ognuno dei cammini attuali e si crea un nuovo cammino per ognuno dei nodi che rispetta tali condizioni, aggiungendolo al cammino per il quale rispetta le condizioni. Un cammino cessa di esistere dal momento che nessun nodo rispetta le condizioni di uno dei passaggi a partire dall’ultimo nodo di quel cammino.

Terminati tutti i passaggi di una regola XPath l’insieme di nodi identificato da quella regola è dato dall’insieme dell’ultimo di ogni cammino.

Ogni passaggio di una regola XPath è dagli altri da un carattere ‘/’, ogni passo corrisponde ad un movimento sull’albero generato dal file XML. Il movimento può essere compiuto in ogni direzione, si può salire verso la radice, e quindi i nodi antenati del nodo corrente, scendere verso le foglie, e quindi i nodi discendenti, e muoversi orizzontalmente verso i fratelli del nodo corrente.

Ogni “passo” è composto da 3 componenti, che stabiliscono condizioni da applicare ai nodi, definiti dal W3C come:

-*Axis*

-*Node test*

-*Predicate*

Ogni passo è rappresentato, in forma estesa, con la seguente sintassi:   
*Axis::Node Test[Predicate]*

***Axis***

L’*axis* determina una condizione sullo spostamento lungo l’albero a partire dal nodo corrente. Il nodo corrente è nel caso del primo passo di una regola il nodo iniziale (o la radice nel caso di un percorso assoluto), mentre nel caso di passi successivi l’ultimo nodo del cammino a cui si sta applicando il corrente passo.

I principali assi consentono:

Lo spostamento verticale, *child* per uno spostamento verso un figlio del nodo corrente e dunque rispettano la condizione solamente i figli nodi del corrente, *descendant* che pone una condizione rispettata da ogni nodo discendente e *descendant-or-self* (abbreviabile con ‘//’) che include anche il nodo corrente, *parent* rispettata dai nodi genitore e *ancestor* di nodi antenato o *ancestor-or-self*  che include anche il nodo corrente.

Lo spostamento orizzontale, *following-sibling* condizione rispettata dai nodi fratelli successivi al nodo corrente e *preceding-sibling* rispettata dai nodi fratelli precedenti.

Lo spostamento nel nodo corrente, *self* (abbreviabile con ‘.’)  pone una condizione verificata solamente dal nodo corrente, dunque per non effettuare uno spostamento e *attribute* (abbreviabile con ‘@’) per ottenere tutti gli attributi del nodo corrente.

**Node Tests**

Il campo *node test* serve a porre una condizione sui nodi. È possibile ad esempio cercare nodi con specifici nomi del *namespace*, come ad esempio cercare per un nodo di tipo *href* oppure è possibile cercare nodi di un *namespace* il cui prefisso sia già stato fissato utilizzando la forma *prefix:name*.

Vi sono poi altri *node test* generici come ‘*\*’* per porre una condizione rispettata da ogni nodo, *text()* rispettata dai nodi di tipo testo e *comment()* valida per i nodi di tipo commento

**Predicate**

Il campo *predicate,* che è omissibile, contiene delle condizioni booleane aggiuntive per la selezione di un nodo. Il nodo su cui il *predicate* viene testato è considerato valido per la regola XPath contenente il *predicate* solamente nel caso il valore restituito sia true. Vi possono essere più *predicate* per ogni “passo”. I *predicate* vengono rappresentati racchiusi da parentesi quadre. I *predicate* vengono confrontati sequenzialmente con ognuno dei nodi attualmente selezionati dal “passo” corrente.

Le espressioni rappresentabili nel *predicate* sono molteplici, come ad esempio a le funzioni booleane come *or, and, !=, =, <, >, <=,>=* oppure *true(),* che restituisce appunto *true,* e *false()* ma anchefunzioni matematiche come ad esempio *mod* per il modulo, funzioni specifiche per le stringhe come ad esempio *contains(string,string)* che restituisce *true* se la seconda stringa è contenuta nella prima o   
*string-length(string)*  che restituisce la dimensione della stringa e funzioni specifiche dei file XML come ad esempio *last()* che restituisce un numero uguale a quello degli elementi selezionati dal “passo” corrente, *position()* che restituisce la posizione del nodo che si sta attualmente confrontando con i *predicate*  e *count(node-set)* che restituisce il numero di nodi presenti nel *node-set* passato come argomento.

**Bibliografia**

Classe, Treccani, Enciclopedia della Matematica (2013), Accesso 01/09/2019

http://www.treccani.it/enciclopedia/classe\_%28Enciclopedia-della-Matematica%29/

Partizione, Treccani, Vocabolario on line, Accesso 01/09/2019

http://www.treccani.it/vocabolario/partizione/

Reticolo, Treccani, Enciclopedia della Matematica (2013), Accesso 02/08/2019

http://www.treccani.it/enciclopedia/reticolo\_%28Enciclopedia-della-Matematica%29/

Teoremi del punto fisso, Treccani, Dizionario di Economia e Finanza (2012), Accesso 29/08/19

http://www.treccani.it/enciclopedia/punto-fisso-teoremi-del\_%28Dizionario-di-Economia-e-Finanza%29/

**Sitografia**

W3C, https://www.w3.org/

Wikipedia, https:// wikipedia.org

1. Teoremi del punto fisso, Treccani, Dizionario di Economia e Finanza (2012), Accesso 29/08/19

   http://www.treccani.it/enciclopedia/punto-fisso-teoremi-del\_%28Dizionario-di-Economia-e-Finanza%29/ [↑](#footnote-ref-1)
2. Figura 4 The original uploader was Tinctorius at English Wikipedia.Later versions were uploaded by EdC at en.wikipedia. [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0/)] [↑](#footnote-ref-2)
3. Classe, Treccani, Enciclopedia della Matematica (2013), Accesso 01/09/2019

   http://www.treccani.it/enciclopedia/classe\_%28Enciclopedia-della-Matematica%29/ [↑](#footnote-ref-3)
4. Partizione, Treccani, Vocabolario on line, Accesso 01/09/2019

   http://www.treccani.it/vocabolario/partizione/ [↑](#footnote-ref-4)
5. Reticolo, Treccani, Enciclopedia della Matematica (2013), Accesso 02/08/2019

   http://www.treccani.it/enciclopedia/reticolo\_%28Enciclopedia-della-Matematica%29/ [↑](#footnote-ref-5)
6. Walber [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)] [↑](#footnote-ref-6)