

Metodi di Ingegneria della Conoscenza applicati alle Homepage delle scuole superiori italiane

# Corso didattico

* Ingegneria della Conoscenza [063507], Facoltà di “Informatica”
* A.A. 2022/23

# Gruppo di lavoro

* Vincenzo Di Bisceglie [745751] [v.dibisceglie3@studenti.uniba.it](mailto:v.dibisceglie3@studenti.uniba.it)

# Repository

* <https://github.com/vodibe/icon-745751>

# Sommario

Corso didattico 1

Gruppo di lavoro 1

Repository 1

Sommario 2

Introduzione 3

Idea del progetto 3

Metriche di usabilità già esistenti 3

Metrica di usabilità adottata per questo progetto 3

Elenco argomenti di interesse 4

Costruzione del ground truth 5

Sommario 5

Strumenti utilizzati 5

Decisioni di Progetto 5

Valutazione 5

Rappresentazione grafo e ricerca soluzioni 6

Modello NaiveDOM 6

Costruzione del Ground Truth: calcolo metrica con algoritmo di ricerca ad-hoc 7

Decisioni di Progetto 7

Valutazione 8

Sezione argomento 1 9

Sommario 9

Strumenti utilizzati 9

Decisioni di Progetto 9

Valutazione 9

Conclusioni 10

Bibliografia 11

# Introduzione

## Idea del progetto

**L’idea di fondo** da cui si è partiti per lo sviluppo di questo progetto è l’applicazione di alcuni metodi di Ingegneria della Conoscenza su un dominio di interesse, in questo caso l’usabilità di una pagina web. Questo richiede che prima si vada a circoscrivere un ambito di riferimento, che nel nostro caso, è l’insieme delle Homepage delle scuole superiori pubbliche italiane (09/2023).

## Metriche di usabilità già esistenti

Le **metriche rilevanti proposte in letteratura** e che potrebbero essere applicate nel contesto di questo progetto sono le Euristiche di Nielsen  [[1]](#b01) le WCAG 2.1  [[2]](#b02) per le quali però gli strumenti software ad essi correlati ([qui elencati](https://www.w3.org/WAI/ER/tools/?q=wcag-21-w3c-web-content-accessibility-guidelines-21)) non sono adatti alla natura di questo progetto perché analizzano il codice sorgente della pagina e danno consigli per rimediare le linee guida non rispettate. Altri strumenti controllano condizioni di accessibilità da parte di utenti con handicap (ad es. controllano che la palette di colori sia accessibile, che supportino l’interazione con hardware ausiliari, …)

Altre metriche rilevanti (SUS Score  [[3]](#b03)) non sono state prese in considerazione perché richiedono un campione di persone alle quali sottoporre un questionario.

## Metrica di usabilità adottata per questo progetto

Pertanto, ai fini del progetto assumeremo che questa **nuova metrica di usabilità** corrisponde a un voto (1☆ - 5☆) che uno studente/docente/genitore (che non ha mai interagito con la Homepage prima d’ora) assegna, tenendo conto di quanto l’interfaccia sia ordinata e funzionale. Approfondiremo questa metrica nelle sezioni seguenti.

# Elenco argomenti di interesse

In questa sezione indichiamo le fasi del progetto e per ciascuna di esse gli argomenti coinvolti:

1. [Costruzione del ground truth](#_Costruzione_del_ground).  
   Poiché all’inizio non disponiamo di una valutazione per tutte le Homepage, ci immedesimiamo in un visitatore della pagina, ne osserviamo gli aspetti grafici e funzionali (in altre parole osserviamo il valore di alcune *feature iniziali*), e diamo una valutazione.
2. Riproduzione del ground truth.  
   La fase 1 è esosa di risorse e quindi cerchiamo ora di riprodurre il ground truth in una forma che sia adatta all’apprendimento automatico. Questa fase si articola in due sotto-fasi:
   1. Osservazione di caratteristiche ottenibili in modo automatico per ciascun sito, mediante **rappresentazione dello spazio di ricerca tramite albero**.
   2. Costruzione + Valutazione di **modelli di apprendimento supervisionato** che, a partire dalle feature secondarie per ciascun sito (individuate al punto 2.1) simulano il giudizio dell’utente.
3. Deduzione di informazioni utili e statistiche.
   1. Costruzione + Valutazione di modelli di apprendimento non supervisionato.
   2. Operazioni di deduzione con ragionamento relazionale e basi di conoscenza.

# Costruzione del ground truth

Si è ipotizzato che in generale, un visitatore osservando la pagina, può valutare il grado di usabilità con una scala [1, 5] con step di 0.1 punti.

* [1, 1.9]: **Sito estremamente caotico**  
  Non esiste un menu; la disposizione di tutti gli elementi è disordinata, per cui è difficile individuare le sezioni che l’utente vuole visitare.
* [2, 2.9]: **Sito confuso** Es. <https://www.maurolicomessina.edu.it/>  
  Esiste un menu; la disposizione di gran parte degli elementi è disordinata e la pagina dà l’impressione di essere troppo lunga.
* [3, 3.9]: **Sito accettabile** Es. <https://www.isii.it/>  
  Esiste un menu che reindirizza il visitatore a gran parte delle sezioni di suo interesse; la pagina però contiene elementi non raggruppati e quindi confusionari.
* [4, 5]: **Sito ordinato** Es. <https://www.liceofermicanosa.edu.it/>  
  Sito accettabile e che inoltre contiene pochi o nessun elemento non raggruppato.

A questo punto, il giudizio dell’utente dipende dalle seguenti features:

|  |  |  |  |
| --- | --- | --- | --- |
| *Feature* | *Descrizione* | *Dominio* | *u.m.* |
| page\_template | Template grafico 1…8=ID del template 9=non segue un template |  |  |
| page\_menu\_or | Orientamento menu 0=non esiste 1=solo orizzontale 2=solo verticale 3=orizzontale e verticale |  |  |
| page\_ungrouped\_multim | Elementi grafici non raggruppati |  |  |

La valutazione è la seguente:

|  |  |  |  |
| --- | --- | --- | --- |
| *Feature* | *Descrizione* | *Dominio* | *u.m.* |
| metric | Valutazione |  |  |

# Rappresentazione grafo e ricerca soluzioni

Moduli Python correlati:

* /agent/webscraper/NaiveDOM.py
* /agent/webscraper/NaiveDOMSearcher.py

## Modello NaiveDOM

Le conoscenze acquisite da questa parte del corso sono servite per introdurre il concetto di NaiveDOM (NDOM) che è un modello DOM semplificato della pagina ottenuto dal parsing del codice sorgente HTML.

Un NDOM è un grafo diretto e pesato, avente struttura ad albero. E’ tale per cui:

* **Ha numero finito di nodi ed è aciclico** (diretta conseguenza del fatto che è un DOM semplificato)
* Ciascun **nodo** è un elemento della pagina, e quindi è identificato univocamente dal suo XPath  [[4]](#b04).

A ciascun nodo sono associati una label (per fini di rappresentazione grafica) e le sue coordinate (x, y).

* Il **nodo radice** è l’XPath del tag <body> della pagina.
* I **nodi interni** sono gli XPath dei tag che contengono potenzialmente, tra i loro discendenti, un testo leggibile. Ad es. <body>, <header>, <section>, <nav> ecc…

Sono esclusi i tag <div>, visto che sono assai frequenti e non semplificano (ma complicano) l’albero finale.

* I **nodi foglia** possono essere di tre tipi:
  + XPath dei tag che non contengono un testo leggibile, ad es. <section>, <img>, ecc…
  + XPath dei tag che contengono sicuramente un testo leggibile, ad es. <a>, <h1>, ecc…
  + Il testo leggibile, a patto che abbia una lunghezza breve.

Per quanto riguarda gli **archi** del NDOM e il loro costo, è necessario prima osservare direttamente un esempio di pagina web e poi fare qualche osservazione.



Figura 1. Homepage di un sito + distanza tra nodo radice del NDOM e suoi figli diretti.

Osservando lo screenshot di una Homepage, notiamo che il nodo radice <body> ha ovviamente coordinate (0,0). I figli diretti di <body> sono gli elementi arancioni e, solo in questo caso, la distanza tra padre e figlio è puramente verticale: questo è ovvio perché l’occhio umano inizia osservando dal basso verso l’alto. In tutti gli altri casi si provvede a calcolare la distanza euclidea.

Il **costo dell’arco** è una funzione della distanza, il cui grafico è descritto di seguito.

Grafico funzione

## Costruzione del Ground Truth: calcolo metrica con algoritmo di ricerca ad-hoc

A questo punto si può descrivere come avviene il calcolo della metrica. Si definisce dapprima un **dizionario di Target**, in cui per ciascuno di essi è presente un ID e una lista di keywords. L’utente trova il Target se, partendo dal nodo radice, esamina la pagina e individua un nodo la cui label contiene una qualunque keyword. Di seguito viene proposto un esempio di dizionario dei Target.

\_TARGETS\_DEFAULT = {

1: ["circolari", "comunicazion"],

2: ["organigramma", "organizzazione"],

3: ["notizie"],

4: ["progett"],

5: ["regolament"],

6: ["amministrazione trasparente"],

7: ["registro"],

8: ["indirizzo", "i luoghi", "dove siamo", "contatti"],

}

L’algoritmo di ricerca proposto gode delle seguenti proprietà:

* Un percorso per il quale l’ultimo nodo ha profondità 0 o 1 (vale a dire, il nodo radice e tutti i percorsi dal nodo radice ai suoi figli diretti) viene aggiunto a una coda con priorità, in cui il percorso a costo minore sarà il primo ad essere esaminato. Questo è ovvio perché una persona passa ad esaminare prima le voci del menu principale rispetto alle voci del footer (che si trovano a fine pagina). Successivamente, gli alberi radicati nei figli diretti della radice vengono esaminati in modalità DFS.
* E’ completo, cioè certo di trovare una soluzione se essa esiste.
* E’ ottimale ??
* Non va in loop (diretta conseguenza della struttura del NDOM)
* Come l’algoritmo DFS, ha complessità di spazio ove è il branching factor e è la profondità del nodo goal; complessità di tempo .

Pertanto, questo suggerisce che la metrica venga calcolata facendo i seguenti step:

1. Applicare l’algoritmo di ricerca sul NDOM per trovare un percorso valido per l’i-esimo Target.
2. Se un tale percorso esiste, calcolare il suo costo totale, altrimenti applicare un costo di default.
3. Assumendo che tutti i Target siano equamente importanti, calcolarne la media dei costi del punto 2.

## Decisioni di Progetto

Tag blaclist

Configurazione dei componenti (e.g. API/librerie) utilizzati, ad es. parametri, soglie, ecc.   
e di eventuali metodi specifici utilizzati a tale scopo

## Valutazione

Paragrafi che richiamino (non spieghino, se standard) le metriche adottate   
+ tabelle sui risultati e loro discussione

# Sezione argomento 1

## Sommario

Un paragrafo che chiarisca la rappresentazione della conoscenza scelta per KB (modelli di ragionamento / apprendimento), dati, BK, ...

## Strumenti utilizzati

Breve: non serve spiegare come funzionano se implementano modelli ben noti   
(basta indicare dei riferimenti bibliografici)

Dilungarsi solo su eventuali modelli/algoritmi originali ideati dal gruppo

## Decisioni di Progetto

Configurazione dei componenti (e.g. API/librerie) utilizzati, ad es. parametri, soglie, ecc.   
e di eventuali metodi specifici utilizzati a tale scopo

## Valutazione

Paragrafi che richiamino (non spieghino, se standard) le metriche adottate   
+ tabelle sui risultati e loro discussione

# Conclusioni

Un paragrafo che riassuma le valutazioni e delinei possibili sviluppi, ad. es. problematiche non affrontate per questioni di tempo (per eventuali estensioni da parte di altri gruppi).

# Bibliografia

|  |  |
| --- | --- |
| [1] | [Online]. Available: https://en.wikipedia.org/wiki/Heuristic\_evaluation#Nielsen's\_heuristics. |
| [2] | [Online]. Available: https://www.w3.org/TR/WCAG21/. |
| [3] | [Online]. Available: https://en.wikipedia.org/wiki/System\_usability\_scale. |
| [4] | [Online]. Available: https://en.wikipedia.org/wiki/XPath. |