

Sistema di valutazione e predizione di usabilità delle Homepage delle scuole superiori italiane

# Corso didattico

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

# Gruppo di lavoro

1. 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 già esistenti e ingegnerizzazione di una nuova 3

Elenco argomenti di interesse 4

Rappresentazione grafo e ricerca soluzioni 5

Modello NaiveDOM 5

Calcolo della metrica (Ground Truth) con algoritmo di ricerca ad-hoc 6

Decisioni di Progetto 7

Valutazione 7

Sezione argomento 1 8

Sommario 8

Strumenti utilizzati 8

Decisioni di Progetto 8

Valutazione 8

Conclusioni 9

Bibliografia 10

# Introduzione

## Idea del progetto

**L’idea di fondo** da cui si è partiti allo sviluppo di questo progetto è la necessità definire un indicatore quantitativo della facilità con cui un utente possa interagire con un sito web. La progettazione di una tale metrica necessita che prima si vada a circoscrivere un ambito di riferimento, che nel nostro caso, è l’insieme delle Homepage dei siti web delle scuole superiori italiane (09/2023). Successivamente, si sono sfruttati vari metodi di Ingegneria della Conoscenza appresi durante il corso per ingegnerizzare una tale metrica ed estrarre informazioni e statistiche oggettive.

Sito ordinato

Sito pessimo

## Metriche già esistenti e ingegnerizzazione di una nuova

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), fermo restando che gran parte degli strumenti software ([qui elencati](https://www.w3.org/WAI/ER/tools/?q=wcag-21-w3c-web-content-accessibility-guidelines-21)) analizzano il codice sorgente della pagina e danno consigli per rimediare le linee guida non rispettate. I restanti strumenti invece, non sono inerenti alla natura di questo progetto perché controllano l’usabilità della pagina 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.

Pertanto, assumeremo che questa **nuova metrica di usabilità** corrisponde a un voto che uno studente/docente/genitore (che non ha mai interagito con la Homepage prima d’ora) assegna, tenendo conto di quanto velocemente è in grado di individuare i suoi contenuti di interesse (ciascuno dei quali chiamato, in questo progetto, **Target**). Questo equivale a fare due considerazioni:

1. Tutte le Homepage considerate per l’ingegnerizzazione della metrica soddisfano a priori tutte le Euristiche di Nielsen, tranne l’euristica #6 “*Recognition rather than recall*”.
2. Inoltre, tali Homepage soddisfano a priori tutte le WCAG 2.1, tranne quelle classificate al punto “[2.4 – Navigable](https://www.w3.org/TR/WCAG21/#navigable)”.

Nei paragrafi successivi questa metrica verrà descritta più approfonditamente.

# Elenco argomenti di interesse

1. Rappresentazione con grafo e ricerca soluzioni
2. Apprendimento supervisionato
3. Clustering

(tratti da sezioni diverse del programma, da indicare esplicitamente)

# 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).

Per ciascun nodo è associato 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 osservare direttamente la pagina web e 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

## Calcolo della metrica (Ground Truth) 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 almeno una delle 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"],

}

Pertanto, questo suggerisce che debba essere progettato anche un algoritmo di ricerca che goda 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. |