Slide 3

TRASPARENZA:

un insieme di meccanismi che permetterebbe a determinati enti (tipicamente la P.A.) di fare trasparenza, ossia di pubblicare verso l'esterno il proprio stato interno.

Esempio: il Comune pubblica come spende il denaro.

Cittadinanza digitale → Il Comune potrebbe comunicare queste informazioni in forma cartacea (attraverso degli avvisi nel paese); la Cittadinanza diventa digitale nel momento in cui questi meccanismi vengono implementati attraverso dei sistemi digitali.

La Cittadinanza è sempre la stessa, diventa digitale nel momento in cui gli strumenti utilizzati diventano informatici.

*Civic accountability* → stabilisce il controllo “sociale” e dovrebbe rendere il rapporto P.A.-cittadino più solido.

Per quanto riguarda il Covid: se il cittadino è informato magari capisce meglio che deve rimanere a casa.

La trasparenza normalmente è un flusso top-down, cioè dall’ente di cui si vuole sapere lo stato vengono emesse delle informazioni che poi arrivano alla cittadinanza.

Slide 4

*Crowdsourcing* → esiste la modalità di condividere un’informazione, in modo che la somma delle parti sia maggiore delle singole parti.

Un gruppo di persone si mette d’accordo ed ognuno di loro fa qualcosa.

Se ognuno condivide le informazioni che ha ricavato, si ottiene un’informazione totale ancor più interessante (si integrano le diverse informazioni).

L’intelligenza collettiva si basa sull’effetto moltiplicatore del *crowdsourcing*.

Slide 6

Tutto ciò che si vuole può essere reso pubblico ed oggetto quindi della trasparenza.

A seconda poi di come si rende disponibile il dato, questo sarà più o meno chiaro (ad esempio la scansione dal cartaceo è pressoché inutile ai fini informatici).

Slide 8

L’opendata è l’implementazione, il meccanismo con cui si fa trasparenza.

È un qualunque dato che viene reso disponibile liberamente e senza vincoli di utilizzo per chiunque.

Slide 11

Sono state date delle classificazioni per valutare sia qualitativamente che quantitativamente un opendata, dal punto di vista della sua accessibilità…

**Berners-Lee** si è inventato le 5 stelle dell’opendata, ossia classifica l’opendata in base al formato e alla licenza.

**Davis** fornisce una classificazione anche in base all’utilità dell’opendata stesso.

Slide 12

**1 stella**: l’OD è in formato non machine-readable e ha una licenza libera (quindi disponibile).

**2 stelle**: formato machine-readable e licenza non libera.

Ad esempio occorre una licenza di Excel per poter elaborare un XLS (foglio elettronico).

**3 stelle**: formato libero e non si è vincolato dall’uso di particolare strumenti per poter elaborare quei dati.

**4/5 stelle**: dati che vengono metadatizzati, formato libero, machine-readable e di semantica convenzionale nota.

Slide 13

Ci sono degli aspetti che Berners-Lee non ha considerato, che poi invece Davis ha ripreso.

Slide 18

Ci sono delle norme di legge che obbligano la P.A. a pubblicare i dati (rendere tarsparenti i dati).

Allora la P.A.lo fa ma “mette i bastoni tra le ruote”.

Slide 19

E lo fa in questo modo:

esistono una serie di tecniche che consentono di fare trasparenza ma senza farla; nel senso limitando i danni della trasparenza.

Si inventano allora gli *webstacles* che sono delle barriere tecnologiche al raggiungimento del dato.

Ossia si mette il dato ma è poco leggibile, ad esempio una scansione dal carteceo, o addirittura si inserisce il CAPTCHA, cioè per poter ottenere l’informazione l’utente deve prima compilare una cosa a mano.

Limitazione QoS → si rende il dato scaricabile, ma si fa impiegare tanto tempo nel download (in modo da far “annoiare” l’utente e indurlo ad abbondare l’attività di scaricamento).

Slide 20

Se si evince che c’è questo meccanismo di webstacles in atto si può tentare di aggirarlo, utilizzando lo

*scraping*.

È un nome generico (per indicare una valanga di too) per consultare un sito web in modo automatico.

Tecniche di *scraping*:

• wget e curl sono due comandi che permettono di scaricare il contenuto di una pagina web al’interno di uno script.

• *browser scripting* → programmare il browser per fargli fare determinate attività

• *headless browsing* → browser senza interfaccia grafica che possono essere programmati

A volte basta leggere l’URL e modificare i parametri per ottenere una determinata informazione.

Slide 22

Ci sono anche siti che commercialmente si occupano di fare *scraping* di altri siti, in modo da rendere facile l’interazione con il sito di cui si intende fare scraping.

Slide 25

Per sapere i dati sull’aria, bisognava compilare il CAPTCHA e arrivava per email lo zip con l’archivio.

Il problema è che bisogna fare circa 20k query col CAPCTHA e ci si impiega anni.

Slide 26

Si può allora mettere in atto uno *scraper* che periodicamente catturi lo stato corrente dell’aria e lo salvi in un dato posto.

Slide 31

Perché ARPA dovrebbe rendere difficile l’acquisizione dei dati (inserendo tutti quei CAPTCHA)?

Per il fatto che si può notare che determinati provvedimenti presi non hanno avuto particolari benefici, ad esempio l’introduzione dell’Area C non ha avuto effetti.

Slide 43

Si tratta di un’elaborazione dei login di un access point, è un grafo.

Sopra ci sono i MAC address dei dispositivi e sotto le date: quindi si vede quali MAC address si sono collegati nelle varie date.

È una visualizzazione intuitiva: subito fa capire cosa vuol far vedere.

Slide 45

La cosa bella di molti siti opendata è che sono accessibili via API.

Ci si può scaricare in locale il dato, ma anche accedervi tramite API.

Le API lavorano tipicamente col paradigma REST e restituiscono XML o JSON (solitamente JSON) come risultato.

Slide 47

Esempi di siti costruiti su dataset pubblici.

Slide 49

Tutto ciò che dice la scienza dovrebbe essere riproducibile, verificabile…

Ma non è sempre così: ci sono dei paper che descrivono protocolli da seguire per ottenere un dato risultato, ma a volte questi risultati non si riescono ad ottenere → quel paper varrà 0!

Il sito indicato tiene traccia dei paper ritirati poiché i risultati descritti nel doumento non sono riproducibili.

Slide 50

Una soluzione a questo problema, almeno a livello del dato, è rendere il dato il più fruibile possibile.

Se un ricercatore ha fatto operazioni su un dato dataset, deve pubblicarlo per poter capire come si è arrivati alla conclusione che espone nel paper.

Lo pubblica tramite un DOI, un link univoco per quel dato dataset → così chi voglia riprodurre i risultati esposti nel suo paper, può utilizzare la stessa versione del dataset.

Se si apportano modifiche al dataset, questo necessita di un nuovo DOI.