

DaCena: Specifica dei requisiti

June 9, 2017

Nome gruppo: SOKE
Numero: 24

Componenti del gruppo:
Angiolillo Andrea 761678
Belingheri Omar 761702
Khayam Adam 761763

Contents

1	Introduzione	3
1.1	Scopo del progetto	3
1.2	Dizionario	4
1.3	Acronimi	5
1.4	References	5
2	Descrizione Generale	7
2.1	Visione del prodotto	7
2.2	Funzioni del software	10
2.3	Caratteristiche degli utenti e scenari di utilizzo	12
2.4	Assunzioni e dipendenze	12
3	Requisiti specifici	13
3.1	Attori del sistema	13
3.2	Diagramma dei casi d'uso	13
3.3	Requisiti interfaccia utente	14
3.4	Requisiti funzionali	14
3.4.1	Effettua registrazione	14
3.4.2	Effettua login	15
3.4.3	Effettua logout	16
3.4.4	Effettua valutazioni associazioni	17
3.4.5	Estrai associazioni articolo	18
3.4.6	Generazione KG	19
3.4.7	Visualizza KG	20
3.4.8	Modifica parametri KG	21
3.4.9	Cerca entità o relazione nel KG	22
3.4.10	Aggiungi articolo	23
3.4.11	Elimina articolo	23
3.4.12	Modifica informazioni utente	24
3.5	Requisiti non funzionali	25
3.5.1	Requisito di performance	25
3.5.2	Attributi di qualità del software	25
3.5.3	Altri requisiti	26

1 Introduzione

Lo scopo di questo documento è quello di fornire una descrizione formale delle specifiche del progetto, nonché il contesto in cui questo si colloca. Inizieremo dando alcune semplici definizioni nella prossima sezione, per poi esplorare più in dettaglio il data-driven journalism e in particolare DaCENA, ovvero il software che andremo a estendere. Successivamente verranno mostrati nel dettaglio tutti i casi d'uso e i requisiti funzionali necessari per l'implementazione di questo lavoro.

Lo standard "IEEE 830-1998", in base al quale è organizzato il presente documento, prevede le seguenti sezioni relative alla definizione dei requisiti: la Sezione 2 dà una descrizione generale del prodotto, dell'ambito in cui verrà collocato, e delle sue funzioni principali, specificando gli obiettivi funzionali di base. In 2.3 verranno introdotte le caratteristiche degli utenti del nostro sistema e in 2.4 sarà introdotto il requisito di indipendenza dalla piattaforma. La Sezione 3 invece specifica più in dettaglio i requisiti (funzionali e non funzionali): in essa si illustrano gli attori del sistema, il diagramma dei casi d'uso completo, si introducono i requisiti dell'interfaccia utente, si elencano i requisiti funzionali e le loro sottofunzionalità, i requisiti non funzionali, e gli attributi di qualità del software (availability e reliability) più altri requisiti come manutenibilità, scalabilità e portabilità.

1.1 Scopo del progetto

Scopo del progetto - Lo scopo di DaCena è quello di permettere una esplorazione innovativa di articoli di giornale attraverso l'utilizzo di knowledge graph definiti sulle entità che compongono l'articolo visualizzato dall'utente. In questo lavoro si è deciso di estendere tale applicativo implementando un algoritmo di online learning to rank che apprenda le preferenze dell'utente in modo da permettere una più efficiente esplorazione e interazione del knowledge graph di un articolo. Successivamente verrà generata una desktop application che affiancherà l'applicazione web di DaCena già esistente. Riassumendo, l'applicazione permetterà a un utente la consultazione efficiente di diversi articoli di giornale attraverso la visualizzazione di un knowledge graph.

Target del sistema - Le funzionalità di DaCena sono state pensate e ideate come supporto al Data-Driven Journalism.

Modalità di utilizzo - Un utente seleziona uno dei diversi articoli di giornale presenti sulla piattaforma, e gli viene mostrato il rispettivo knowledge graph. Questo conterrà le entità contenute nell'articolo, collegate da archi che rappresentano relazioni che sussistono tra di esse (queste informazioni vengono estratte da DBpedia). Successivamente il sistema chiede all'utente di valutare alcune delle relazioni (o associazioni) mostrate e, in base alle preferenze espresse dall'utente, rigenera knowledge graph in modo da evidenziare entità che l'utente potrebbe trovare più interessanti.

Una volta che il grafo è stato visualizzato, l'utente può decidere di valutare ulteriori associazioni per permettere al sistema di raffinare la scelta delle entità

da mostrare.

1.2 Dizionario

Nome	Descrizione
Ontologia	Definizione formale dei nomi, tipi, proprietà e relazioni che esistono tra un insieme di entità. Un'ontologia rappresenta la conoscenza relativa a un certo dominio applicativo
Entità	Oggetto (astratto o concreto) rappresentato in un'ontologia. Può avere diverse proprietà ed essere in relazione con altre entità
Associazione	Indica come due o più relazioni sono collegate tra di loro
Entropia	Misura che indica quanto un elemento è incerto
Relazione	Legame che esiste tra due entità
Tassonomia	Struttura gerarchica di superclassi e sottoclassi tra le entità definite
Tripla	$\langle S, P, O \rangle$ è una tripla in cui S, il soggetto, è legato all'oggetto O tramite la relazione (o proprietà) P
Knowledge Graph	Grafo i cui nodi rappresentano delle entità e i cui archi rappresentano relazioni tra esse
DaCENA	Web application che, scelto un articolo, ne rappresenta le entità principali attraverso un Knowledge Graph
Open Linked Data	Dati strutturati pubblicati liberamente, interrogabili tramite query semantiche

Data-Driven Journalism	Metodologia di stesura articoli di giornale basato sull'analisi e filtraggio di dati di grandi dimensione per la creazione di notizie
Learning to Rank	Insieme di algoritmi di apprendimento che hanno lo scopo di ordinare una lista di elementi presi in input secondo le loro caratteristiche apprese durante la fase di training
Knowledge Exploration	Applicazione di tecniche di machine learning, tipicamente supervisionate, semi- supervisionate o per rinforzo, nella costruzione di modelli di ranking in ambiti come l'information retrieval
DBpedia	DBpedia è un progetto che ha come scopo l'estrazione di informazioni strutturate a partire dai dati che sono presenti su Wikipedia

1.3 Acronimi

Nome	Acronimo
KG	Knowledge Graph
DaCena	Data Context for News Articles
OLD	Open Linked Data
DDJ	Data-Driven Journalism
LR	Learning to Rank
KE	Knowledge Exploration

1.4 References

1. Federico Bianchi "Active learning to rank su associazioni semantiche interessanti per un utente", tesi (2016)
2. Roberto Zerbo "Valutazione qualitativa e quantitativa dell'applicazione DaCENA",

3. Nikos Bikakis and Timos Sellis. Exploration and visualization in the web of big linked data: A survey of the state of the art. arXiv preprint arXiv:1601.08059, 2016.
4. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
5. Teh, Yee Whye. "Dirichlet process." Encyclopedia of machine learning. Springer US, 2011. 280-287
6. LI Hang. A short introduction to learning to rank. IEICE TRANSACTIONS on Information and Systems, 94(10):1854–1862, 2011.

2 Descrizione Generale

2.1 Visione del prodotto

Contesto applicativo - Con l'espressione "Data-driven journalism" (DDJ) si fa riferimento a un fenomeno che trae le sue origini negli anni '70 (con il "Precision Journalism"), ma che solo nel 2009 si è diffuso con questo nome. Essa descrive un nuovo tipo di giornalismo fortemente basato sullo studio e l'analisi di grandi sorgenti di dati, reso possibile dalla sempre crescente disponibilità di dati liberamente fruibili online. Nel data-driven journalism si impiegano nuove tecnologie per semplificare, e allo stesso tempo potenziare, l'attività giornalistica: questo porta a una maggiore rapidità nella preparazione delle notizie, utilizzando informazioni filtrate da fonti diverse e recuperate via web.

Il processo si può pensare come organizzato in quattro fasi principali

1. Ricerca: estrazione di dati da una o più fonti di diverso tipo (es. Dati governativi, KG...);
2. Filtraggio: si procede ad eliminare tutti quei dati che risultano superflui o poco utili agli scopi dell'applicazione. In questa fase i dati vengono trasformati in modo da essere raccolti in maniera strutturata e quindi più semplice da analizzare;
3. Visualizzazione: rende possibile visualizzare i dati che sono stati raccolti e organizzati;
4. Pubblicazione.



Figure 1: Processo DDJ

Il DDJ beneficia in maniera sostanziale dalla presenza (e continua diffusione ed espansione) degli Open Linked Data: al crescere del volume di dati gratuitamente fruibili cresce conseguentemente la quantità/qualità dei dati a disposizione del giornalista.

È in questo contesto che si inserisce DaCENA, il cui scopo è quello di supportare i giornalisti nell'orientarsi nella vastità di dati presenti online, estraendoli, organizzandoli e visualizzandoli in modo automatico.

L'applicativo DaCena - DaCENA è un software che consente a un utente di consultare un articolo di giornale attraverso il suo knowledge graph (KG). Un KG è un grafo i cui nodi rappresentano delle entità e i cui archi rappresentano relazioni tra esse; in altre parole è un modo di organizzare la conoscenza relativa un certo dominio applicativo. Le entità sono a loro volta descritte in delle ontologie, ovvero delle strutture che definiscono in maniera formale i tipi, le proprietà e le possibili relazioni tra entità.

Prima di poter entrare nello specifico delle funzionalità di DaCENA, ci sembra opportuno chiarificare i concetti appena introdotti mediante un esempio di come una associazione viene visualizzata su un KG.



Figure 2: Esempio di associazione: per come questa associazione è definita nella sua ontologia, essa deve legare un oggetto di tipo “presidente” a un oggetto di tipo “Paese”, che in questo caso specifico si realizzano in Obama e con Stati Uniti

DaCENA è quindi un tool che costruisce un KG contenente le entità principali riconosciute nel testo in input (solitamente un articolo di giornale) e trovate in DBpedia, organizzandole secondo le relazioni che sussistono tra esse e consentendo all'utente di riorganizzare e modificare il grafo.

Una volta ricevuto il testo in input, DaCENA procede con la fase di “Entity linking”, ovvero il riconoscimento delle entità presenti nel testo. Durante questa attività è fondamentale che DaCENA riesca a disambiguare i termini, per esempio collegando correttamente la parola “Paris” all'entità che rappresenta la città di Parigi e non a quella di Paris Hilton. Successivamente al riconoscimento delle entità inizia la fase di generazione delle associazioni svolta attraverso DBpedia. La fase finale consiste nella generazione e visualizzazione del KG dell'articolo.

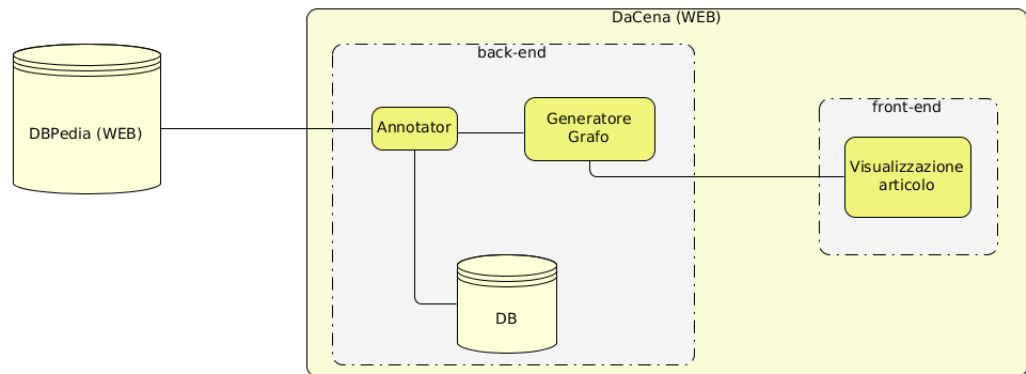


Figure 3: Sistema as-is: Il sistema al momento

Online Learning - Questo lavoro ha il compito di ridefinire le fasi dell'applicativo DaCena al fine di incorporare nel sistema un algoritmo di online learning che permetta di mettere in risalto nel KG le entità che sono più interessanti per il bisogno informativo dell'utente.

Successivamente alle fasi di pre-processing dei dati, già presenti nell'applicativo, segue la fase di learning svolta nel seguente modo:

1. Un algoritmo di dirichlet clustering si occupa di selezionare le associazioni più rappresentative dei rispettivi cluster generati (centroidi);
2. Viene chiesto all'utente che sta visualizzando l'articolo di valutare queste associazioni con uno score compreso tra 1 e 6 (dove 1 indica un basso interesse e 6 alto interesse);
3. L'algoritmo di online learning si occupa di ordinare le restanti associazioni in base alle valutazioni che ha dato l'utente;
4. Viene generato il grafo;
5. Se l'utente è soddisfatto della visualizzazione grafica allora la fase di learning termina, altrimenti può valutare ulteriori due associazioni;
6. La fase di selezione delle ulteriori associazioni da far valutare viene svolta basandosi sulla misura di entropia: le due associazioni che hanno entropia maggiore vengono selezionate;

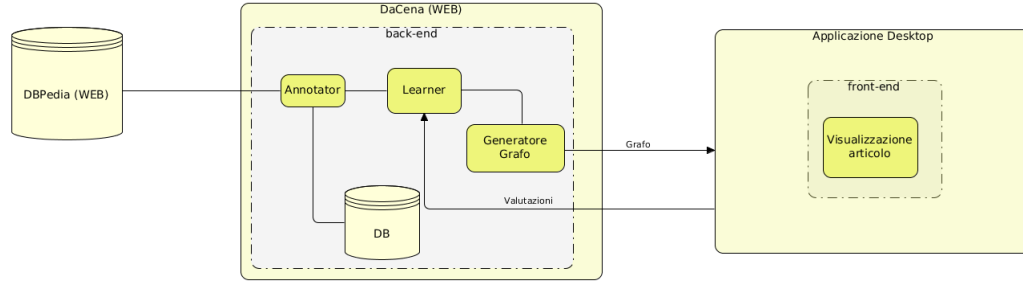


Figure 4: Sistema to-be

7. Si ritorna al punto 2.

Desktop Application - Come si può vedere dalla Figura 4, oltre alla parte di learner, verrà definita un'applicazione desktop che permetta all'utente di visualizzare l'articolo e il KG a esso associato. Naturalmente in questa fase non deve e non viene definita alcuna architettura del sistema, la Figure 4 indica soltanto uno schema ad alto livello degli elementi che fanno già parte o che faranno parte di DaCena, la loro collocazione non è definitiva e verrà trattata nelle fasi successive.

2.2 Funzioni del software

Obiettivi funzionali di base - Le funzionalità basilari del sistema sono l'estrazione delle entità presenti in un articolo e il reperimento di entità e proprietà aggiuntive da DBpedia per permettere una maggiore comprensione di come si relazionano tra di loro. La fase di reperimento di entità non menzionate nell'articolo viene svolta per cercare di apportare maggiore conoscenza all'utente. Successivamente verranno definite le relazioni tra le entità appena identificate al fine di generare delle associazioni. Queste informazioni saranno successivamente organizzate in una struttura dati per facilitarne l'utilizzo nei moduli successivi.

Associazioni generate - Un esempio di struttura dati per le associazioni che verrà utilizzata per generare il KG è la seguente:

Source	First_property	Middle_one	Second_property	Destination
Bernie_Sanders	R:birthPlace	New_York	L:region	Hillary_Clinton
Bernie_Sanders	R:otherParty			Democratic_Party

Table 2: In questo esempio sono rappresentate due associazioni, una per ogni riga della tabella

Le colonne Source, Middle_one e Destination indicano i nomi delle entità che verranno rappresentate sul KG. Le relazioni tra queste entità sono rappresentate dalle colonne First_property e Second_Property nel seguente modo: R:birthPlace indica che la relazione birthplace è una freccia che parte dall'entità di sinistra Bernie_Sanders all'entità di destra New_York; il ragionamento inverso viene fatto con L:region.

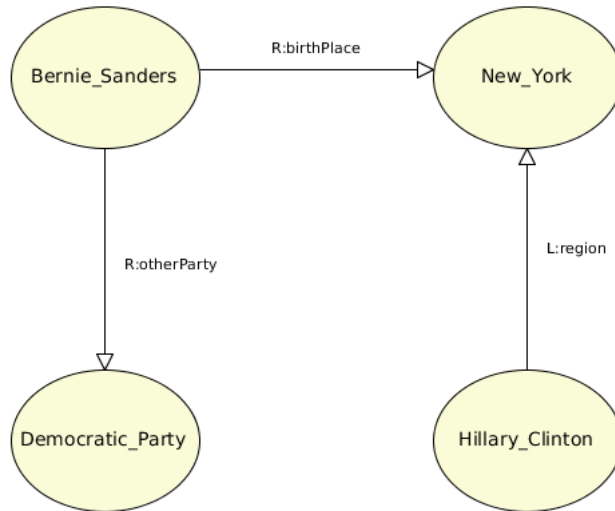


Figure 5: Le associazioni nella tabella 2 rappresentate su KG

2.3 Caratteristiche degli utenti e scenari di utilizzo

Utenti principali del sistema - L'utente del nostro sistema sarà un giornalista/blogger, o una persona che si informa consultando articoli online molto frequentemente. A fronte di questo si suppone che abbia già delle conoscenze pregresse sull'articolo consultato e voglia aumentare la sua conoscenza attraverso la consultazione del KG. Sarà presente un amministratore che gestirà gli articoli e gli utenti che potranno accedere alla piattaforma.

Scenari di utilizzo - Per meglio comprendere come l'utente giornalista e l'amministratore interagiscono con le funzionalità offerte dal sistema, di seguito sono specificati due possibili scenari di utilizzo.

Scenario I: Un utente decide di accedere alla piattaforma utilizzando le sue credenziali di accesso. L'utente sceglierà tra il catalogo di articoli disponibili sulla piattaforma un articolo da consultare. Successivamente verranno mostrate delle associazioni dell'articolo che l'utente valuterà in base alle sue necessità informative. DaCena visualizzerà il KG dell'articolo mettendo in risalto le entità e le relazioni più rilevanti per l'utente in questione. Una volta visualizzato il KG l'utente può decidere di valutare ulteriori associazioni per permettere alla piattaforma di apprendere ulteriormente le sue preferenze e in questo caso DaCena aggiornerà il KG con la nuova predizione.

Scenario II: L'utente amministratore accede alla piattaforma e gli viene mostrata una schermata con le seguenti azioni disponibili:

1. Aggiungi/Modifica/Rimuovi articolo: in questo caso l'amministratore effettua delle modifiche sul catalogo degli articoli che verranno mostrate agli utenti una volta acceduti all'applicazione;
2. Modifica informazioni utente: l'amministrazione modifica l'email e/o la password di un utente.

2.4 Assunzioni e dipendenze

Indipendenza da piattaforma - L'idea alla base del progetto è quella di creare un sistema, diffuso tramite internet, sfruttabile gratuitamente da chiunque. Questa caratteristica è possibile rendendo l'applicazione indipendente da piattaforma, cioè deve poter funzionare su qualsiasi macchina a prescindere dalla sua configurazione software/hardware.

Testing dell'indipendenza - L'applicazione verrà eseguita su piattaforma Windows, MacOS e Linux. Il funzionamento di DaCena dipende, ovviamente, dalla disponibilità di un collegamento ad internet.

Online Learning - Si assume che gli algoritmi di online learning riescano a comprendere le reali preferenze dell'utente su un determinato argomento.

3 Requisiti specifici

3.1 Attori del sistema

Attori del sistema:

1. **Utente:** utente autenticato che utilizza il sistema per leggere un articolo;
2. **Admin:** utente che può aggiungere/rimuovere articoli di giornale;
3. **Guest:** utente non autenticato che può fare solo il login;
4. **DBpedia:** piattaforma online che fornisce le relazioni tra le entità.

3.2 Diagramma dei casi d'uso

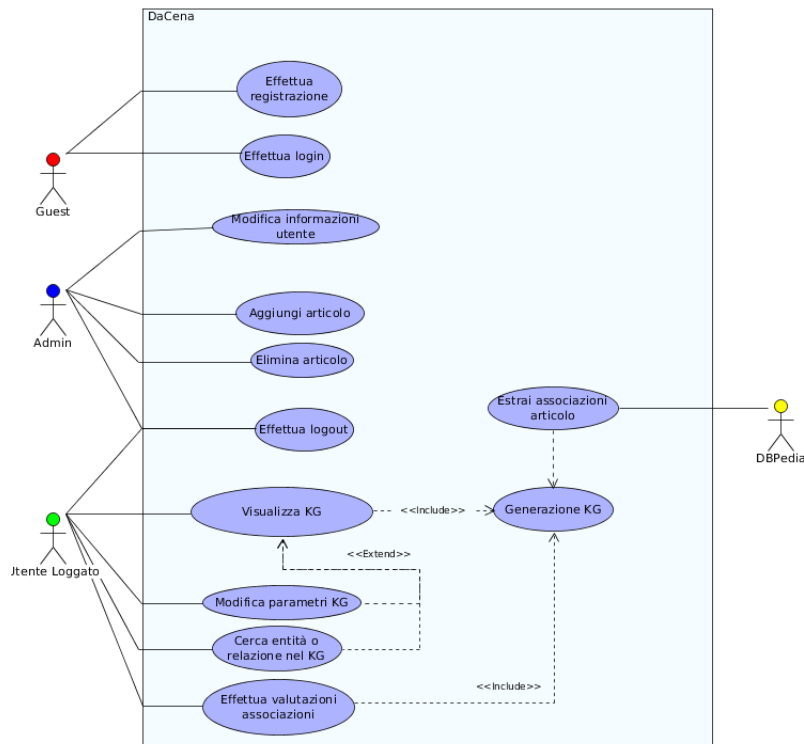


Figure 6: Diagramma dei casi d'uso

3.3 Requisiti interfaccia utente

Funzionalità interfaccia - A seguire le funzionalità, esposte in dettaglio, dell'interfaccia utente. Questa dovrà permettere di:

- Effettuare la registrazione di un nuovo utente;
- Effettuare il login/logout;
- Selezionare un articolo precedentemente caricato dall'amministratore sul server;
- Avviare il processo di apprendimento valutando delle associazioni;
- Informare l'utente se sono stati aggiunti nuovi articoli dalla sua ultima visita;
- Visualizzare il KG;
- Permettere la modifica dei parametri del KG;
- Permettere la ricerca di una entità o relazione sul KG;
- Permettere all'utente amministratore di aggiungere/rimuovere nuovi articoli;
- Permettere all'utente amministratore di modificare le informazioni utente.

3.4 Requisiti funzionali

3.4.1 Effettua registrazione

ID: 01_reg.

Attori primari: Guest.

Parti interessate ed interessi: L'utente Guest vuole poter creare un account personale per accedere all'applicazione.

Input:

- Email valida;
- Password valida;

Pre-condizioni: L'utente Guest è un utente in possesso di una email attiva e di una password valida.

Post-condizioni: L'utente Guest viene promosso a utente del sistema.

Scenario principale di successo:

1. Guest inserisce l'indirizzo email e password;
2. Guest conferma i dati inseriti;
3. Indirizzo email e password vengono inviati al server;

4. Il server risponde positivamente generando un account e garantendo all'utente l'accesso al servizio.

Estensioni:

1a. La password non è in formato conforme allo standard (per complessità o altro)

1. la registrazione fallisce;
2. l'utente deve scegliere un'altra password;

3a. L'email non è in formato conforme allo standard

1. la registrazione fallisce;
2. l'utente deve inserire un'altra email;

Elenco delle varianti tecnologiche e dei dati: Connessione a internet.

Frequenza di ripetizione: BASSA.

3.4.2 Effettua login

ID: 02_login.

Attori primari: Guest.

Parti interessate ed interessi: L'utente Guest vuole poter effettuare il login al servizio offerto dall'applicazione.

Input:

- Email valida;
- Password valida;

Pre-condizioni: L'utente Guest è un utente in possesso di una email registrata e di una password valida.

Post-condizioni: L'utente Guest viene promosso a utente del sistema.

Scenario principale di successo:

1. Guest inserisce l'indirizzo email e password;
2. Guest conferma i dati inseriti;
3. Indirizzo email e password vengono inviati al server;
4. Il server risponde positivamente garantendogli l'accesso al servizio

Estensioni:

3a. L'email non è riconosciuta come registrata

1. il login fallisce;
2. il server richiede di inserire nuovamente i dati.

3b. La password e email non coincidono

1. il login fallisce;
2. il server richiede di inserire i dati.

Elenco delle varianti tecnologiche e dei dati: Connessione a internet.

Frequenza di ripetizione: BASSA.

3.4.3 Effettua logout

ID: 03_logout.

Attori primari: Utente.

Input: -

Parti interessate ed interessi: L'utente vuole poter uscire dall'applicazione.

Pre-condizioni: L'utente è loggato

Post-condizioni: L'utente non è più loggato

Scenario principale di successo

1. L'utente clicca il tasto logout;
2. Il server risponde positivamente garantendogli l'uscita dal servizio.

Estensioni: -

Frequenza di ripetizione: BASSA.

3.4.4 Effettua valutazioni associazioni

ID: 04_val.

Attori primari: Utente.

Parti interessate ed interessi: L'utente deve poter indicare il suo livello di interesse rispetto a certe associazioni, opportunamente selezionate, appartenenti all'articolo

Input: valore intero compreso tra 1 e 6 che indica una valutazione dell'associazione (1 = poco rilevante, 6 = molto rilevante).

Pre-condizioni:

L'utente:

1. è loggato nell'applicazione;
2. ha selezionato un articolo.

Post-condizioni: L'applicazione salva le valutazioni dell'utente e avvia la fase di learning.

Scenario principale di successo:

1. Il server seleziona delle associazioni da far valutare all'utente
2. L'utente valuta le associazioni.

Estensioni:

2.a L'utente inserisce una valutazione non valida (ad esempio un carattere alfabtico): viene restituito un messaggio di errore.

Frequenza di ripetizione: ALTA.

3.4.5 Estrai associazioni articolo

ID: 5_assoc.

Attori primari: DBpedia

Parti interessate ed interessi: -

Input: Corpo articolo.

Pre-condizioni: L'articolo è presente nel sistema.

Post-condizioni: Le entità presenti nell'articolo selezionato e le associazioni che le collegano vengono salvate.

Scenario principale di successo:

1. Il server si collega a DBpedia per estrarre le entità e le associazioni che compongono l'articolo selezionato;
2. Le entità e le associazioni vengono salvate in un apposito datastore.

Estensioni: -

Frequenza di ripetizione: MEDIA.

3.4.6 Generazione KG

ID: 06_KG.

Attori primari: -

Parti interessate ed interessi: -

Input: entità, associazioni

Pre-condizioni:

L'utente:

1. E' entrato sulla piattaforma;
2. Ha selezionato un articolo;

Post-condizioni: Viene generato il KG associato all'articolo.

Scenario principale di successo:

1. Viene eseguito il modulo online learning sull'intero insieme di associazioni dell'articolo;
2. Il KG viene generato.

Estensioni:

- 1a. Se è la prima volta che l'utente apre l'articolo, e non ha ancora valutato associazioni, allora:
- 2a. Viene eseguito il clustering sulle associazioni del grafo
- 3a. Si scelgono delle associazioni come centroidi
- 4a. Il KG viene generato

Frequenza di ripetizione: ALTA.

3.4.7 Visualizza KG

ID: 07_Visualiz.

Attori primari: Utente.

Parti interessate ed interessi: L'utente vuole visualizzare il KG.

Input: entità e associazioni del KG

Pre-condizioni:

L'utente:

1. è entrato nell'applicazione;
2. ha selezionato un articolo;

Il sistema ha generato il KG associato all'articolo selezionato;

Post-condizioni: Viene visualizzato il KG associato all'articolo.

Scenario principale di successo:

1. Il server invia il KG generato al client;
2. L'utente vede il KG di fianco all'articolo e lo può esplorare (usandone le funzioni di ricerca, ecc.).

Estensioni: -

Frequenza di ripetizione: ALTA.

3.4.8 Modifica parametri KG

ID: 08_Setting_graph.

Attori primari: Utente.

Parti interessate ed interessi: L'utente può decidere di modificare i seguenti parametri del KG:

1. Degree of collapsing: parametro k che permette di mostrare soltanto le entità collegate con almeno k relazioni;
2. Gravity: parametro j che indica la distanza (in cm) tra le entità del grafo.

Input: parametro k e/o j.

Pre-condizioni:

L'utente:

1. è entrato nell'applicazione;
2. ha selezionato un articolo;
3. ha visualizzato il KG.

Il sistema ha generato il KG associato all'articolo selezionato.

Post-condizioni:

- Vengono mostrate soltanto le entità con un degree of collapsing maggiore o uguale a k.
e/o
- Viene impostata j come distanza (in cm) tra le entità che formano il grafo.

Scenario principale di successo:

- Degree of collapsing:
 1. L'utente che visualizza il grafo è interessato a visualizzare soltanto determinate entità con almeno un certo grado k di interconnessione;
 2. L'utente imposta il valore di k;
 3. Viene aggiornato il grafo.
- Gravity:
 1. l'utente vuole modificare le distanze tra i nodi del grafo;
 2. L'utente imposta il valore di j;
 3. Viene aggiornato il grafo.

Estensioni:

2a. Nel caso in cui il valore K sia un valore non possibile viene restituito un messaggio di errore e il grafo non viene alterato.

2b. Nel caso in cui il valore J sia un valore non possibile viene restituito un messaggio di errore e il grafo non viene alterato.

Frequenza di ripetizione: ALTA.

3.4.9 Cerca entità o relazione nel KG

ID: 09_Search.

Attori primari: Utente.

Parti interessate ed interessi: L'utente vuole cercare una entità o una relazione di interesse nel KG.

Input: Stringa che indica il nome dell'entità o della relazione.

Pre-condizioni:

L'utente:

1. è entrato nell'applicazione;
2. ha selezionato un articolo;
3. ha visualizzato il KG.

Il sistema ha generato il KG associato all'articolo selezionato.

Post-condizioni: Viene evidenziata l'entità o la relazione cercata.

Scenario principale di successo:

1. L'utente inserisce nella apposita textbox il nome dell'entità o della relazione;
2. Viene evidenziata l'entità o la relazione desiderata.

Estensioni:

2a. viene inserito un nome non presente nel KG: non viene evidenziata nessuna entità.

Frequenza di ripetizione: ALTA.

3.4.10 Aggiungi articolo

ID:10_add_article.

Attori primari: Amministratore.

Parti interessate ed interessi: L'amministratore vuole aggiungere un nuovo articolo alla piattaforma.

Input: Articolo.

Pre-condizioni: L'amministratore è entrato nel sistema.

Post-condizioni: Viene aggiunto un nuovo articolo consultabile nel sistema.

Scenario principale di successo:

1. L'amministratore entra nella apposita sezione dedicata agli articoli;
2. Inserisce un nuovo articolo (titolo, autore, corpo, data, immagine);
3. Viene generato un nuovo articolo consultabile da tutti gli utenti sulla piattaforma.

Estensioni:

3a: I dati inseriti non sono conformi (ad esempio manca il titolo): viene restituito un messaggio di errore.

Frequenza di ripetizione: BASSA.

3.4.11 Elimina articolo

ID:11_delete_article.

Attori primari: Amministratore.

Parti interessate ed interessi: L'amministratore vuole eliminare un articolo dalla piattaforma.

Input: Articolo.

Pre-condizioni: L'amministratore è entrato nel sistema;

Post-condizioni: Viene eliminato un articolo.

Scenario principale di successo:

1. L'amministratore entra nella apposita sezione dedicata agli articoli;
2. Cerca l'articolo;
3. Seleziona l'opzione elimina articolo;
4. Il sistema elimina l'articolo selezionato.

Estensioni: -

Frequenza di ripetizione: BASSA.

3.4.12 Modifica informazioni utente

ID: 12_user_information.

Attori primari: Amministratore, Utente.

Parti interessate ed interessi: L'amministratore vuole modificare l'email e/o la password di un utente registrato sulla piattaforma.

Input: email e/o password.

Pre-condizioni:

1. L'amministratore è entrato nel sistema;
2. L'utente deve essere un utente registrato che ha dimenticato la password o vuole cambiare email;
3. L'utente ha contattato il supporto.

Post-condizioni: Vengono modificate le credenziali dell'utente.

Scenario principale di successo:

1. L'amministratore entra nella apposita sezione dedicata agli utenti;
2. Cerca l'utente;
3. Seleziona l'opzione "Modifica credenziali utente";
4. L'amministratore inserisce nella apposita textbox la nuova email e/o password;
5. Il sistema notifica l'utente con una email.

Estensioni:

4a. Email e/o password non validi: in questo caso viene restituito un messaggio di errore.

Frequenza di ripetizione: BASSA.

3.5 Requisiti non funzionali

3.5.1 Requisito di performance

Collo di bottiglia per la velocità - Considerata la natura del progetto le performance sono estremamente variabili: dipendono dalla potenza della macchina host e dalla velocità della rete utilizzata per l'accesso al sistema. A fronte di questo non possono essere formulati requisiti precisi sulle performance, ma si deve assicurare che una volta lanciata, l'applicazione visualizzi il KG in un tempo finito.

Requisiti minimi di performance - Nonostante l'impossibilità di definire requisiti precisi di performance vengono comunque definiti dei limiti minimi supponendo la disponibilità di una connessione internet e di un calcolatore di fascia media:

- la generazione e visualizzazione del KG deve avvenire in massimo 2 secondi;
- le modifiche dei parametri del KG devono essere visibili in massimo 1 secondo;
- la selezione dell'entità ricercata dall'utente nel KG deve essere eseguita in 0.5 secondi.

3.5.2 Attributi di qualità del software

Availability e Reliability - Durante lo sviluppo, contrariamente a come generalmente accade nello sviluppo di sistemi software, si dovrà porre principalmente il focus non sulla reliability ma sulla availability del sistema. È più tollerabile che DaCena restituisca classificazioni errate rispetto alla non risposta causata da un blocco del sistema, in quanto la consultazione del KG da parte dell'utente deve essere sempre possibile.

Soluzione possibile per garantire availability - Una possibile soluzione per garantire availability è quella di suddividere in moduli distinti la parte di apprendimento da quella di generazione del KG così da consentirne una sua visualizzazione anche in caso di errore del modulo di learning. Questa soluzione deve essere accompagnata da una gestione efficace delle eccezioni possibili.

Soluzione possibile per garantire reliability - Per garantire un adeguato livello di reliability è stato pensato di utilizzare un metodo di recovery che, in caso di interruzione non prevista dell'esecuzione del sistema o logout, permetta di riprendere il procedimento dallo stato in cui il sistema si trovava prima di quell'evento.

3.5.3 Altri requisiti

Manutenibilità e scalabilità - Questi requisiti vengono garantiti strutturando DaCena in modo modulare. Essi sono di fondamentale importanza nel caso in cui il sistema dovesse crescere e quindi venire esteso.

Portabilità - DaCena deve essere portabile, ossia indipendente da ogni piattaforma hardware e software, come già definito nella Sezione 2.4 - *Assunzioni e dipendenze*. Il soddisfacimento di questo requisito dovrà essere garantito sia dal linguaggio di programmazione che verrà scelto per l'implementazione finale, sia dalla tecnologia che verrà scelta per implementare la piattaforma desktop.