Corso di Laurea in Informatica

TESI DI LAUREA

TooDY

Tool for Detecting variabilitY

Relatore
Laura Semini

Candidato
Matteo Giorgi

ANNO ACCADEMICO 2022/2023

Indice

Sommario

Questo studio si inserisce nel contesto della ricerca sulla definizione di *Linee di Prodotto Software* basate su documenti di requisiti redatti in linguaggio naturale. Le ambiguità presenti in tali documenti possono generare disallineamenti tra le aspettative del cliente e il prodotto finale, causando spesso revisioni non previste degli artefatti. Si è notato che un'espressione ambigua può talvolta servire come strumento per posticipare decisioni.

Ampliando questa prospettiva, studi precedenti hanno evidenziato come l'identificazione dell'ambiguità possa rivelare elementi di variabilità nascosti nei requisiti, ricorrendo a specifici marcatori di variabilità che si discostano dai consueti indicatori di ambiguità.

Nella presente tesi, sulla base di un prototipo esistente, si è sviluppato uno strumento di elaborazione linguistica in grado di rilevare tali indicatori di variabilità nei documenti di requisiti. Questo strumento, dotato di capacità di analisi lessicale e sintattica, si avvale della libreria *spaCy* per la sua realizzazione.

Capitolo 1

Introduzione

L'abilità di sviluppare software di qualità ha inizio con la definizione precisa ed esaustiva dei requisiti che si richiede al sistema di soddisfare. Il documento dei requisiti, in tal senso, rappresenta l'elemento base su cui si fonda l'intero ciclo di vita del progetto software. La redazione di tale documento tuttavia, non è compito semplice e ciò è principalmente dovuto alla natura ambigua ed intrinsecamente imprecisa del linguaggio naturale comunemente utilizzato per esprimerlo. In un progetto software la formalizzazione dei requisiti è dunque pratica essenziale e, se non opportunamente curata, possibile veicolo di ambiguità, che spesso portano a interpretazioni errate, incoerenze ed eventuali fallimenti nel soddisfare le aspettative del cliente.

Il presente lavoro di tesi si colloca in questo scenario, proponendo lo sviluppo di *TooDY* (tool for detecting variability): una web-app focalizzata sull'identificazione della variabilità nei documenti dei requisiti redatti in linguaggio naturale. L'obiettivo è quello di creare uno strumento facile e veloce da usare; un supporto concreto nell'analisi dei requisiti, utile a mitigare i problemi legati all'ambiguità linguistica e ridurre il divario tra le aspettative del cliente e il prodotto software sviluppato.

1.1 Strumenti utilizzati

Nell'ambito di questo progetto, sono stati utilizzati diversi strumenti che hanno facilitato lo sviluppo e la realizzazione del tool; tra questi, meritano una menzione particolare *Flask* e *spaCy*. *Flask* è un micro framework per interfacce web, estremamente flessibile e leggero, che ha permesso di sviluppare un server dedicato all'elaborazione delle richieste relative all'analisi dei documenti dei requisiti e alla gestione di un da-

tabase contenente lo storico di ciascun utente registrato al servizio web: l'utente può quindi editare e verificare i documenti caricati, verificandone la correttezza formale e controllandone i vari casi di ambiguità e variabilità. *spaCy*, invece, è una libreria open source di elaborazione del linguaggio naturale che ha fornito le funzionalità di analisi lessicale e sintattica necessarie per l'individuazione degli indicatori di variabilità nei documenti analizzati. La scelta di tali strumenti è stata dettata dalla necessità di avere un sistema flessibile, efficiente e in grado di gestire complesse analisi linguistiche in modo efficace.

Unitamente, l'interfaccia intuitiva della web-app facilita l'interazione con il sistema, permettendo agli utenti di gestire i documenti direttamente sulla piattaforma ed eseguire l'analisi in modo semplice e diretto: la possibilità di visualizzare in tempo reale i risultati dell'analisi è certamente uno strumento utile per le fasi di verifica e la validazione, promuovendo così un approccio più accurato e riflessivo nella definizione dei requisiti del software.

1.2 Motivazioni

La necessità di una maggiore formalizzazione e di nuovi strumenti che potessero assistere in modo efficace l'analisi dei requisiti ha rappresentato uno stimolo alla realizzazione di *TooDY*.

L'idea alla base del progetto è quella di trasformare un termine ambiguo da ostacolo a risorsa e utilizzarlo come mezzo per rimandare alcune decisioni; al contempo, identificare aspetti nascosti di variabilità nei requisiti.

Le richieste iniziali del progetto delineavano la necessità di una piattaforma eclettica, facilmente accessibile attraverso diversi sistemi operativi e questo requisito ha orientato la scelta verso lo sviluppo di una web-app, permettendo così un accesso ubiquo e una fruibilità trasversale indipendentemente dal sistema utilizzato. L'interfaccia così realizzata, non solo facilita l'interazione con il sistema di analisi, ma si evolve in una piattaforma efficiente, comprensiva di gestione dei documenti, fornendo all'utente uno strumento completo per l'elaborazione dei testi.

1.3 Struttura e contenuto

La presente dissertazione è organizzata seguendo una logica chiara e progressiva, mirata a fornire una comprensione completa del lavoro svolto e degli obiettivi raggiunti con lo sviluppo di *TooDY*.

Il capitolo *SPLE & Paradigmi*, approfondisce i vari paradigmi legati alle linee di prodotto software, esplorando metodologie come *FAST*, *RSEB* e *FODA*. In particolare, si discutono le tecniche di analisi del dominio e la rappresentazione attraverso il diagramma delle features. Viene anche affrontata la tematica dell'ambiguità linguistica e come essa può introdurre variabilità nei requisiti.

In *Tecniche di NLP*, si esplorano le fondamenta dell'elaborazione del linguaggio naturale. Vengono presentate le principali tecniche, come *Tokenizzazione*, *POS-Tagging* e *Parsing*, illustrando come queste siano state applicate nel contesto del progetto. Vengono inoltre descritte le principali librerie utilizzate, con un focus particolare su *spaCy*.

Il cuore della tesi è rappresentato da, *Studio, progettazione e lavoro svolto*, in cui si entra nel dettaglio tecnico del progetto *TooDY*. Vengono presentati il materiale di partenza, l'architettura del server *Flask* e l'implementazione del parser. Questa sezione illustra concretamente come le teorie e le tecniche discusse nei capitoli precedenti siano state applicate nella realizzazione pratica del tool.

Infine, si riflette sulle competenze acquisite, sulle potenzialità di *TooDY* e sugli sviluppi futuri, offrendo una visione d'insieme delle implicazioni e delle prospettive del progetto.

Concludendo, l'elaborato intende fornire un contributo nell'ambito della ricerca relativa alla specifica di *Linee di Prodotto Software* a partire da documenti di requisiti in linguaggio naturale. Attraverso l'implementazione di *TooDY*, si punta a fornire uno strumento utile ed efficace per l'identificazione della variabilità, con l'obiettivo di migliorare la qualità del processo di analisi dei requisiti e del software prodotto.

Capitolo 2

Software Product Line engineering

Una Software Product Line (SPL) rappresenta un'approccio strategico alla ingegneria del software che permette di creare una varietà di prodotti software correlati da un insieme comune di risorse, garantendo al contempo che ciascun prodotto soddisfi i requisiti specifici del cliente. Questo approccio si basa sull'idea di capitalizzare su aspetti comuni e variabili tra i diversi prodotti software per raggiungere economie di scala, ridurre i tempi di sviluppo e migliorare la qualità del software. L'obiettivo è sviluppare una famiglia di prodotti che condividono una struttura comune, pur avendo variazioni per soddisfare le esigenze di mercati o clienti diversi.

La Software Product Line Engineering (SPLE), invece, è la disciplina ingegneristica che guida l'organizzazione, la creazione e il mantenimento di una SPL. Si concentra sulla definizione e l'implementazione di processi e metodi robusti per gestire la variabilità e l'ereditarietà all'interno della linea di prodotti, garantendo che ogni prodotto generato dalla linea di prodotti soddisfi i requisiti di qualità e funzionalità. Attraverso la SPLE, le organizzazioni possono formalizzare e gestire in modo efficace l'evoluzione della linea di prodotti, dalla concezione alla consegna e al mantenimento. La SPLE enfatizza la creazione di un'architettura comune e di un insieme di componenti riusabili, oltre alla definizione di processi per configurare e assemblare questi componenti in prodotti individuali.

Al cuore della SPLE vi è la gestione della variabilità, che si riferisce alla capacità di un sistema di essere efficientemente estendibile, modificabile, personalizzabile e configurabile per differenti contesti. La gestione

della variabilità permette di catturare e gestire le differenze e le similitudini tra i prodotti della linea in modo sistematico, permettendo una produzione efficiente e controllata di vari membri della linea di prodotti.

2.1 FAST

2.2 RSEB

2.3 FODA

2.4 Ambiguità e Variabilità

Capitolo 3

Tecniche di NLP

3.1 Tokenizzazione

3.2 POS-Tagging

3.3 Parsing

3.4 Analisi semantica

3.5 Librerie e dizionari per NLP

Capitolo 4

Studio, progettazione e lavoro svolto

```
@app.route('/new-backup-code', methods=['GET'])
@login_required
def new_backup_code():
    backup_code = generate_random_string(8)
    current_user.backup_code = backup_code
    db.session.commit()

    flash(f'Your new backup code is {backup_code}. ' +
          'Keep it in a safe place!', 'info')
    return redirect(url_for('main'))
```

4.1 Materiale di partenza

4.2 Architettura e implementazione server *Flask*

4.3 Implementazione parser

Capitolo 5

Conclusioni

5.1 Familiarità tecniche usate e competenze acquisite

5.2 Sviluppi futuri e riflessioni critiche