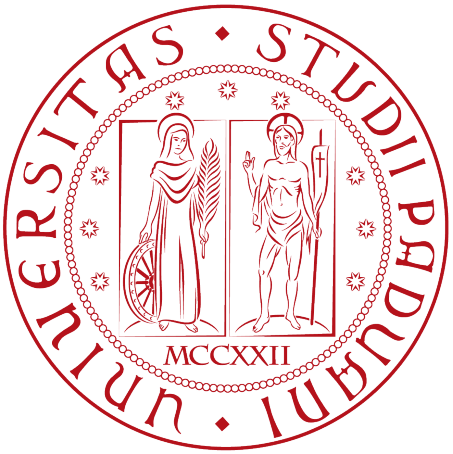
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Analisi e sviluppo di un'applicazione per la configurazione automatica di una chatbot professionale

Tesi di laurea triennale

Relatore
Prof.Ballan Lamberto

Laureando
Stefano Zanatta

Anno Accademico 2018-2019

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

— Oscar Wilde

Dedicato a ...

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di trecento ore, dal laureando Stefano Zanatta presso l'azienda PAT s.r.l.Gli obbiettivi principali del progetto erano:

* studio del motore semantico di Engagent, la chatbot professionale dell'azienda;
* studio dei risultati di un algoritmo di clustering, sviluppato da ricercatori esterni all'azienda;
* integrazione automatica dei cluster con Engagent, eseguendo operazioni di pos-tagging.

"Life is really simple, but we insist on making it complicated"

— Confucius

Ringraziamenti

Innanzitutto, vorrei ringrazione il Prof. Lamberto Ballan, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura della tesi.
Vorrei ringraziare il tutor aziendale Davide Bastianetto, per avermi dato l'opportunità di svolgere lo stage a PAT.
Desidero ringraziare con affetto i miei genitori e mio fratello per il sostegno morale (ed economico) dimostratomi durante questi anni.

Padova, Settembre 2019 Stefano Zanatta

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 L'azienda

PAT s.r.lè un'azienda italiana che da 25 anni sviluppa soluzioni software per altre aziende e privati.
L'azienda lavora su 5 diversi macro-progetti, uno dei quali è Engagent, la chatbot professionale oggetto dei miei due mesi di stage.
Dal 2013, PAT è entrata a far parte di Zucchetti Group.

1.1.1 Prodotti e servizi

L'azienda offre ai suoi clienti l'automatizzazione dei processi e il miglioramento dell'*user experience* dei loro prodotti. PAT concretizza quesi obbiettivi attraverso i seguenti prodotti:

* Engagent: chatbot semi-automatizzata per uso professionale;
* Helpdesk;
* Infinite: *CRM* software orientato alla relazione tra cliente e azienda;
* Brain *Interactive*: piattaforma per governare dei servizi personalizzati attraverso dei diagrammi di flusso;
* Teammee: piattaforma per la comunicazione tra i dipendenti in un'azienda, usando la logica dei social networks;

Engagent

Engagent è una chatbot orientata al business, con un agente virtuale integrato.
In *backend*, un motore semantico permette di capire cosa sta chiedendo l'utente e trovare la risposta più coerente. Se la domanda è troppo complessa, il motore semantico estrate la categoria della domanda e la reinderizza all'operatore (umano) adeguato.

1.1.2 Organizzazione e Metodo di lavoro

L'azienda è divisa in più gruppi di lavoro, uno per ogni macro-progetto, oltre alla segreteria e direzione.

Ogni team è separato dagli altri, anche se la collaborazione tra le parti è necessaria.
Tutti i team di sviluppo in PAT s.r.lseguono una metodologia Agile. Questa metodologia fa parte delle metodologie iterative. Le brevi iterazioni (o sprint, di circa 3-4 settimane) sono seguite dalla *review* del lavoro svolto. Il focus principale si trova nel cliente, difatti ci deve essere una interazione costante per capire quali sono le *feature* più importanti, ovvero quelli da sviluppare prima.
Il team a cui ho preso parte applica questa metodologia. Il contatto con il cliente è frequente, che sia manutenzione o nuove *features* da sviluppare. La piccola dimensione del team e le riunioni giornaliere permettono una buona collaborazione. Il team è gestito da un responsabile che organizza le riunioni e comunica con il manager dell'azienda.
Per quanto mi rigurarda, ho adottato senza difficoltà queste metodologie, perché molto simili a quelle utilizzate durante il progetto di ingegneria del software.

1.2 Strumenti e Tecnologie

1.2.1 Ambiente di lavoro

L'ambiente di lavoro utilizzato è Windows 10, assieme al pacchetto office per la maggior parte delle attività.
I team di sviluppo hanno libera scelta sugli editor. Ho scelto Visual studio, per la sua flessibilità.
Ogni sviluppatore ha a disposizione un PC fisso e un portatile per le riunioni.

1.2.2 Test

I test vengno eseguiti su più livelli:

* **test di Engagent:** test di accettazione e di sistema. Viene verificato che il motore semantico funzioni correttamente;
* **Postman:** test di integrazione. Permette di generare delle richieste http alle API sviluppate;
* **pytest:** test di unità e integrazione. Ambiente di test specifico di Python. Tramite pytest-cov, permette di calcolare il code coverage;
* **pylint:** analisi statica del codice. Test statici per Python;

1.3 Organizzazione del testo

Il primo capitolo contiene una panoramica dell' azienda e le tecnologie utilizzate PAT s.r.l;

Il secondo capitolo contiene la pianificazione progetto;

Il terzo capitolo approfondisce nel dettaglio i requisiti del progetto, descrivendo il processo di analisi che ha portato alla loro stesura;

Il quarto capitolo approfondisce l'architettura del software sviluppato, con il supporto di esempi specifici e schemi ad alto livello;

Il quinto capitolo approfondisce le tecniche di verifica e validazione utilizzate durante lo stage, con i relativi risultati;

Nel sesto capitolo contiene il resoconto dello stage.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

* termini di uso non comune vengono definiti nel glossario, situato alla fine del documento;
* per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*[g];
* i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Descrizione dello stage

2.1 Il progetto

Il progetto è nato dalla necessità dell'azienda PAT s.r.ldi automatizzare il processo di configurazione del motore semantico di *Engagent*[g], il quale richiede la creazione manuale di *synset*[g]e *regole*[g]. Questo processo è economicamente fattibile e vantaggioso finché le dimensioni delle regole create sono ridotte, ma il costo cresce esponenzialmente con l'aumentare delle regole.
Più regole rendono più precisa la chatbot, ma incrementano di conseguenza i [g]synset da inserire, prolungando i tempi di compilazione del'*NLP*[g]da qualche giorno a settimane.
Per risolvere questo problema, PAT s.r.lha scomposto il processo nei seguenti *task* da automatizzare:

1. creazione delle *regole*[g];
2. creazione dei synset;
3. raffinamento dei risultati;
4. creazione del file di configurazione *NLP* per il motore semantico;

2.1.1 Creazione delle regole

Questo task è il più difficile da automatizzare, perché richiede la definizione di *match* contenenti categorie correlate tra loro. Inoltre, non esistono delle regole uguali per tutti, ma ogni settore ha regole diverse.
La soluzione è stata trovata nell'intelligenza artificiale, più in particolare nel clustering. Tramite l'analisi di *chat* e *FAQs* archiviate, è possibile generare delle regole allo stato grezzo.
Il problema è la bassa affidabilità dei risultati. Possibili soluzioni:

- più dati in input (poco realizzabile nel breve periodo);
- algoritmo più complesso (in lavorazione);
- raffinamento manuale dei risultati (anche se manuale, richiede meno tempo di creare l'NLP da zero, soluzione migliore nel breve termine);

* raffinamento automatico dei risultati (tramite *POS-tagging*) (buon compromesso tra il raffinamento manuale e le altre due soluzioni).

2.1.2 Creazione dei synset

La creazione dei *synset* è facilmente automatizzabile (se le regole sono già state create), in quanto basta trovare i sinonimi delle categorie.

2.1.3 Raffinamento dei risultati

I risultati dell'algoritmo di clustering devono essere ripuliti da *stop-words* e regole prive di significato.

2.1.4 creazione dell'NLP

Adattamento dell'output dell'algoritmo di clustering al motore semantico di Engagent.

2.2 Obbiettivi Aziendali

L'obbiettivo di automatizzare il processo di configurazione di Engagent non è nato con il progetto di stage, ma qualche anno fa, mentre l'*IA* diventava sempre più popolare. L'azienda attribuì questo compito a un team esterno. Il loro compito consisteva nel sviluppare un algoritmo di intelligenza artificiale, per la generazione di cluster contenenti gli ingredienti essenziali alla configurazione del motore semantico.
Verso l'inizio dell'anno, i progressi fatti da questo team erano convincenti, quindi PAT s.r.laveva l'intenzione di sperimentare l'integrazione di tali risultati con il proprio sistema.

2.3 Obbiettivi personali

Durante la ricerca dell'azienda per lo stage, volevo contribuire a un progetto software in ambito professionale, facendo contemporaneamente i primi passi nel mondo delle intelligenze artificiali.
Il progetto proposto da PAT racchiudeva queste prerogative: sarei stato inserito in un progetto maturo e, con l'aiuto di esperti nel settore, avrei potuto lavorare con degli algoritmi di clustering.

2.4 Pianificazione

Con l'aiuto del tutor aziendale ho redatto il piano di lavoro, che comprende 300 ore distribuite in 8 ore al giorno, per 5 giorni alla settimana.
La pianificazione ha avuto delle modifiche durante l'avanzare del progetto, vista la sua natura "sperimentale". Per esempio, il linguaggio di programmazione Python è stato accordato assieme al tutor aziendale solamente dopo un'analisi approfondita del problema. Di seguito viene riportata l'ultima versione del piano di lavoro.

* **I settimana:** studio della piattaforma *Engagent*;
* **II settimana:** analisi e stesura del report riguradante il problema descritto in 2.1;

* **III settimana:** ricerca e sperimentazione di possibili soluzioni già esistenti per automatizzare la generazione di sinonimi; analisi e progettazione dell' applicazione NLP-Generator;
* **IV settimana:** preparazione dell'ambiente di lavoro; codifica di NLP-Generator; stesura di test di unità;
* **V settimana:** codifica e miglioramento delle prestazioni di *NLP-Generator*; verfica dei risultati di *NLP-generator* da parte del tutor aziendale
* **VI settimana:** analisi sul miglioramento dei risultati di *NLP-Generator* e codifica; documentazione;
* **VII settimana:** documentazione e validazione;
* **VIII settimana:** collaudo.

2.5 Analisi dei rischi

Assieme al tutor aziendale abbiamo individuato alcuni rischi a cui si potrà andare incontro. Per ognuno è stata trovata una soluzione:
1. Conflitti durante il testing e la modifica in Engagent
Descrizione: La piattaforma Engagent mette a disposizione un ambiente di testing, per provare le configurazioni prima di portarle in produzione. Se due persone lavorano sullo stesso *NLP* è possibile che si generino dei conflitti dannosi all'interno del software..
Soluzione: A ogni membro del team è stato assegnato un dominio su cui lavorare (un sotto-ambiente di testing) per evitare conflitti.

Capitolo 3

Analisi dei requisiti

3.1 Casi d'uso

La progettazione dell'applicazione è iniziata con la stesura dei casi d'uso, supportati da diagrammi dei casi d'uso coerenti con lo standard UML.
I casi d'uso sono aumentati durante tutta la durata dello stage, in quanto il tutor aziendale richiedeva nuove funzionalità del programma.

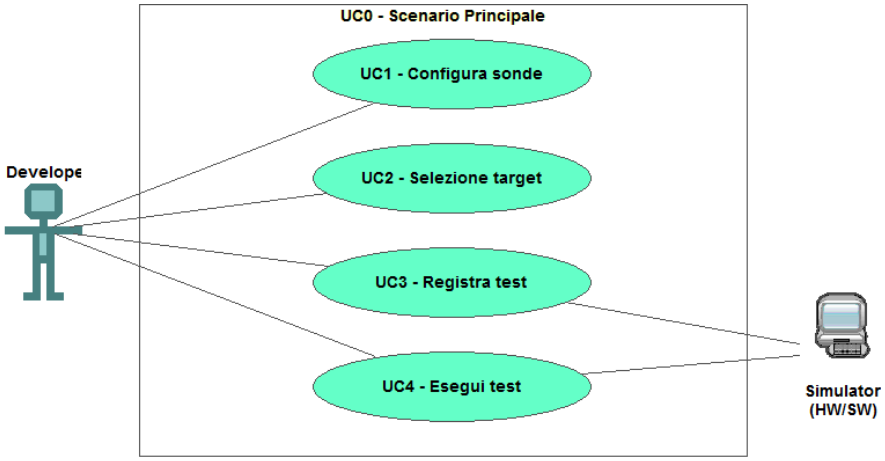


Figura 3.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

3.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.
Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.
Il codice dei requisiti è così strutturato R(F/Q/V)(N/D/O) dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle 3.1, 3.2 e 3.3 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Tabella 3.1: Tabella del tracciamento dei requisti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

Tabella 3.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

Tabella 3.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

Capitolo 4

Progettazione e codifica

Breve introduzione al capitolo

4.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

4.2 Ciclo di vita del software

4.3 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

4.4 Design Pattern utilizzati

4.5 Codifica

Capitolo 5

Verifica e validazione

Capitolo 6

Conclusioni

6.1 Consuntivo finale

6.2 Raggiungimento degli obiettivi

6.3 Conoscenze acquisite

6.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia