

# COLLETTA: PIATTAFORMA PER RACCOLTA DATI MEDIANTE ESERCIZI DI GRAMMATICA

Giulio Paci

2 Novembre 2018



<http://www.mivoq.it/>

info@mivoq.it MIVOQ S.R.L. +39 049 0998335

## Indice

<b>1 Obiettivo del progetto</b>	<b>1</b>
<b>2 Piattaforma di raccolta dati</b>	<b>1</b>
2.1 Insegnanti . . . . .	1
2.2 Allievi . . . . .	2
2.3 Sviluppatori . . . . .	3
2.4 Obiettivi obbligatori . . . . .	3
2.5 Obiettivi desiderabili . . . . .	3
2.6 Obiettivi opzionali . . . . .	3
<b>3 Apprendimento automatico supervisionato</b>	<b>4</b>
3.1 Addestramento ed esecuzione . . . . .	4
3.2 Part-of-Speech tagging . . . . .	4
<b>4 Specifiche di progetto e riferimenti</b>	<b>5</b>
<b>5 Scelte tecnologiche</b>	<b>6</b>
<b>A Note sul capitolato d'appalto</b>	<b>7</b>
A.1 Aspettative della proponente . . . . .	7
A.2 Contatti . . . . .	7
<b>B Note sulla proponente</b>	<b>7</b>
<b>Riferimenti bibliografici</b>	<b>7</b>

# 1 Obiettivo del progetto

L'obiettivo del progetto è realizzare una piattaforma collaborativa di raccolta dati in cui gli utenti possano predisporre e/o svolgere piccoli esercizi di grammatica (per esempio esercizi di analisi grammaticale) e i dati raccolti siano relativi sia agli esercizi predisposti che al loro svolgimento da parte degli utenti. I dati raccolti devono essere utilizzabili da sviluppatori e ricercatori al fine di insegnare ad un elaboratore a svolgere i medesimi esercizi mediante tecniche di apprendimento automatico supervisionato (vedi sezione 3).

Gli attori principali sono quindi utenti che vogliono far pratica di grammatica svolgendo esercizi (allievi), utenti che vogliono predisporre esercizi (insegnanti) e persone interessate ad ottenere i dati delle interazioni, per migliorare il servizio stesso o per scopi di ricerca (sviluppatori).

Allo scopo di chiarire meglio gli obiettivi del capitolato, nelle sezioni che seguono si farà sempre riferimento al compito dell'analisi grammaticale e si forniranno esempi specifici di possibili esercizi. Ciò nonostante non è obbligatorio che la piattaforma implementi esattamente gli esercizi descritti, né che gli esercizi siano prettamente di analisi grammaticale. Eventuali proposte alternative sono gradite, e potranno essere concordate con la proponente.

## 2 Piattaforma di raccolta dati

Il progetto prevede la realizzazione di una piattaforma di raccolta dati. La raccolta dati non deve avvenire in modo esplicito, piuttosto gli utenti della piattaforma devono trovare una utilità intrinseca nel fornire i dati.

Per questo si suggerisce di implementare una piattaforma per preparare ed eseguire semplici esercizi di grammatica, come l'analisi grammaticale. I tre attori principali della piattaforma saranno quindi:

- insegnanti: gli insegnanti hanno interesse a preparare in modo semplice e rapido degli esercizi per i propri allievi;
- allievi: gli allievi hanno interesse ad eseguire gli esercizi ed ottenere un risultato immediato;
- sviluppatori: gli sviluppatori hanno interesse ad accedere ai dati raccolti mediante la piattaforma per automatizzare il servizio.

### 2.1 Insegnanti

L'insegnante che utilizzerà la piattaforma dovrà poter predisporre esercizi di analisi grammaticale in modo comodo. In particolare dovrà poter inserire nuove frasi nel sistema, etichettarle in automatico con informazioni relative all'analisi grammaticale e dovrà poterle correggere, al fine di garantire che i propri allievi ricevano del materiale controllato, sia per quanto riguarda il contenuto delle frasi di partenza che per quanto riguarda il risultato degli esercizi.

Al fine di agevolare il lavoro di preparazione degli esercizi è opportuno utilizzare un software in grado di svolgere gli esercizi impiegando tecniche di apprendimento automatico supervisionato (per esempio Hunpos [Mivoq(2014-2018)] o FreeLing [TALP Research Center, UPC(2008-2018a)]). Poiché questo tipo di prodotto è sempre affetto da un certo margine di errore, l'insegnante dovrà poter correggere i risultati prodotti dal software.

In questo modo la fase di preparazione di un esercizio prevederà:

- inserimento di una frase da parte dell'insegnante;
- esecuzione dell'esercizio da parte di un software, al fine di popolare i dati con cui poter correggere gli esercizi fatti dagli allievi;
- correzione dei dati prodotti dal software da parte dell'insegnante.

La piattaforma dovrà essere in grado di immagazzinare sia i dati grezzi inseriti dall'insegnante (in questo caso le frasi), che i risultati degli esercizi svolti dal software, che le successive correzioni da parte dell'insegnante. Le modifiche dovranno essere tutte immagazzinate e la storia delle modifiche deve poter essere ricostruita a partire dai dati immagazzinati. Allo stesso tempo per l'insegnante dovrà essere possibile accedere all'ultima versione, che verrà utilizzata per predisporre gli esercizi.

Nel caso la stessa frase venga inserita e/o corretta da più insegnanti è opportuno che il sistema sia in grado di ricondurre ogni frase e ogni correzione a ciascun insegnante. Il sistema non dovrà quindi assumere che per ogni singola frase si debba per forza avere una versione unica all'interno della piattaforma. E due insegnanti diversi potranno decidere soluzioni indipendenti.

Allo stesso modo sarebbe opportuno prevedere per uno stesso insegnante la possibilità di fornire più soluzioni alternative anche per una stessa frase. Infatti non sempre per la stessa frase esiste una sola soluzione per l'analisi grammaticale, poiché questa può variare col significato delle parole e la frase può essere ambigua (per es., nella frase "Una vecchia legge la regola." in cui "legge" e "regola" possono essere sia sostantivi che verbi a seconda del significato della frase).

Quando un'insegnante inserisce una frase, potrebbe non volerla condividere pubblicamente, per vari motivi. In questo caso sarebbe opportuno prevedere un apposito sistema per consentire agli insegnanti di monitorare chi può accedere o meno ad una determinata frase o un determinato esercizio.

L'insegnante potrebbe altresì voler predisporre esercizi in varie lingue. In questo caso l'insegnante dovrebbe poter specificare la lingua di una frase e la piattaforma dovrebbe consentire di predisporre esercizi secondo le regole della lingua specificata.

## 2.2 Allievi

L'allievo che accede al sistema dovrà poter svolgere gli esercizi predisposti dall'insegnante e ricevere una valutazione immediata.

PAROLA	ANALISI	
Il	<input type="text"/>	0
bambino	<ul style="list-style-type: none"> <li>aggettivo</li> <li>articolo</li> <li>avverbio</li> <li>congiunzione</li> <li>nome proprio</li> <li>preposizione</li> <li>pronome</li> <li>punteggiatura</li> <li>sostantivo</li> <li>verbo</li> </ul>	0
possiede		0
molti		0
libri		0
di	<input type="text"/>	0
favole	<input type="text"/>	0
pochi	<input type="text"/>	0
di	<input type="text"/>	0
storia	<input type="text"/>	0
.	<input type="text"/>	0
<b>PUNTEGGIO TOTALE:</b>		<b>0</b>

Figura 1: Esercizio di analisi grammaticale [HT Tecnologie Umanistiche(2012-2018)]

In figura 1 è illustrato una possibile implementazione di esercizio di analisi grammaticale, così come realizzato dalla piattaforma Analisi Grammaticale [HT Tecnologie Umanistiche(2012-2018)]. In questo caso all'allievo viene data la possibilità di scegliere le classi grammaticali per ciascuna parola attraverso appositi menu a tendina. Al termine l'allievo riceve una valutazione.

Nella piattaforma di questo progetto, l'allievo dovrà poter esprimere una preferenza per i propri insegnanti. In modo da poter privilegiare le versioni di un insegnante rispetto ad un altro. L'allievo dovrebbe poter effettuare esercizi sia partendo da un elenco di frasi proposte, sia inserendo autonomamente una frase nel sistema. Per ogni frase all'utente dovrebbe essere data la possibilità di selezionare l'insegnante che ha predisposto l'esercizio. Se nessun insegnante ha predisposto un esercizio con quella frase, il sistema automatico (lo stesso utilizzato per predisporre gli esercizi da parte dell'insegnante) potrà comunque essere utilizzato per effettuare una valutazione.

In questo modo lo svolgimento di un esercizio prevederà:

- ricerca dell'esercizio (o attraverso frasi suggerite o attraverso l'inserimento di una frase libera);

- selezione dello specifico esercizio su cui si verrà valutati (quindi selezione dell'insegnante che ha predisposto l'esercizio o dell'algoritmo automatico);
- svolgimento dell'esercizio;
- valutazione.

Anche nel caso degli allievi i dati dovranno essere immagazzinati e contrassegnati opportunamente. Quando si hanno a disposizione gli esercizi effettuati da più utenti, è possibile cercare di dedurre la soluzione più corretta dal comportamento della maggioranza.

Al fine di incentivare l'utente a svolgere esercizi è consigliato fornire all'allievo un cruscotto per visualizzare i propri progressi nel tempo, e prevedere di ricompensarlo periodicamente con il raggiungimento di traguardi (per es., il primo esercizio svolto, il primo esercizio svolto con successo, 5 esercizi consecutivi svolti con successo, ...).

## 2.3 Sviluppatori

Gli sviluppatori sono interessati prevalentemente ad accedere ai dati raccolti dagli utenti al fine di utilizzarli nella fase di addestramento (vedi sezione 3.1) di sistemi di apprendimento automatico. In questo caso lo sviluppatore è generalmente interessato ad ottenere più di una versione dell'annotazione di ogni frase, al fine di poter dedurre quale sia l'annotazione più corretta. Non è raro che in questo ambito gli esseri umani non siano concordi nell'etichettare la medesima frase, anche quando si parla di esperti. Questo può avvenire sia per divergenze di pensiero nella definizione di una certa classe grammaticale (per esempio alcuni potrebbero distinguere l'articolo indeterminativo "un, uno, una" rispetto al numerale, altri potrebbero non distinguere le due classi), sia per ambiguità linguistiche (Per esempio "Portacelo!" può essere derivata sia da "Porta+ci+lo", quindi "portalo a noi", che da "Porta+ce+lo", quindi "portalo là"). Per gli sviluppatori è quindi importante poter conoscere anche lo storico dei dati, in modo da poter estrarre solo i dati di interesse (Per esempio escludendo le correzioni di alcuni utenti, o utilizzando solo le correzioni più frequenti).

Per alcuni sviluppatori potrebbe essere interessante accedere ai modelli realizzati a partire dai dati (vedi sezione 3.1) ed utilizzabili per lo svolgimento automatico degli esercizi.

## 2.4 Obiettivi obbligatori

Gli obiettivi obbligatori del progetto sono:

- realizzazione di una piattaforma per la raccolta dati
  - possibilità di eseguire automaticamente un esercizio (per es. analisi grammaticale) mediante un software basato sull'apprendimento automatico supervisionato;
  - possibilità di correggere l'output automatico e di salvare il risultato al fine di predisporre esercizi rapidamente;
  - possibilità di scaricare i dati collezionati per poter addestrare (vedi sezione 3.1) il software di apprendimento automatico;
- documentazione dell'applicazione
  - analisi dei requisiti;
  - descrizione tecnica.

## 2.5 Obiettivi desiderabili

- possibilità di predisporre e svolgere esercizi, e raccogliere dati in più lingue;
- possibilità di personalizzare i dati prima di scaricarli;
- possibilità di differenziare fra dati pubblici e dati privati.

## 2.6 Obiettivi opzionali

- possibilità di salvare e di scaricare l'intera cronologia delle modifiche ai dati;
- possibilità di creare e scaricare modelli (vedi sezione 3.1) direttamente dalla piattaforma.

### 3 Apprendimento automatico supervisionato

L'apprendimento automatico supervisionato mira a istruire un sistema informatico a risolvere alcuni compiti in maniera automatica. L'apprendimento avviene sulla base di esempi reali di input e output attesi; il risultato è una funzione in grado di avvicinarsi all'output atteso sia per i dati forniti come esempio, sia per quelli non forniti.

Sistemi di questo tipo sono molto diffusi e utilizzati pressoché in ogni ambito, basti pensare al riconoscimento vocale, alla traduzione automatica, ai correttori ortografici, ma anche ai filtri anti spam, ai sistemi di sblocco sulla base di informazioni biometriche, all'analisi delle immagini in medicina. . .

Il presupposto fondamentale per poter utilizzare questo tipo di tecnica è che il compito da svolgere possa essere appreso mediante esempi ed il suo impiego è conveniente qualora non si riesca a codificare un'insieme di regole preciso per il compito in questione e/o siano già disponibili numerosi esempi. Nel caso in cui non siano disponibili dati di esempio, la tendenza odierna è comunque quella di privilegiare questo tipo di tecnica in virtù dei costi: normalmente è più dispendioso codificare delle regole opportune, poiché richiede la disponibilità di esperti sia del settore specifico che di informatica, anziché produrre tali esempi mediante raccolta dati. La raccolta dati può avvenire in modo esplicito, chiedendo alle persone di creare il materiale secondo determinate specifiche (per esempio il progetto VoxForge [VoxForge(2006-2018)] per la raccolta di audio trascritto), o implicito, realizzando un servizio che permetta la raccolta dati (per esempio Google Translate [Google(2006-2018)], in cui le correzioni degli utenti al servizio di traduzione vengono immagazzinate per migliorare il servizio stesso).

In questo capitolo si richiede la messa a punto di una piattaforma il cui scopo ultimo sia la raccolta dati. La raccolta dati deve essere effettuata in modo implicito, cioè deve avvenire attraverso l'utilizzo di un servizio (svolgimento di esercizi). La piattaforma dovrà pubblicizzare lo scopo della raccolta dati e, a differenza di quanto avviene nelle piattaforme proprietarie, consentire l'accesso ai dati raccolti.

#### 3.1 Addestramento ed esecuzione

Ogni sistema di apprendimento automatico prevede due fasi, generalmente realizzate attraverso software separati: la fase di addestramento e quella di esecuzione.

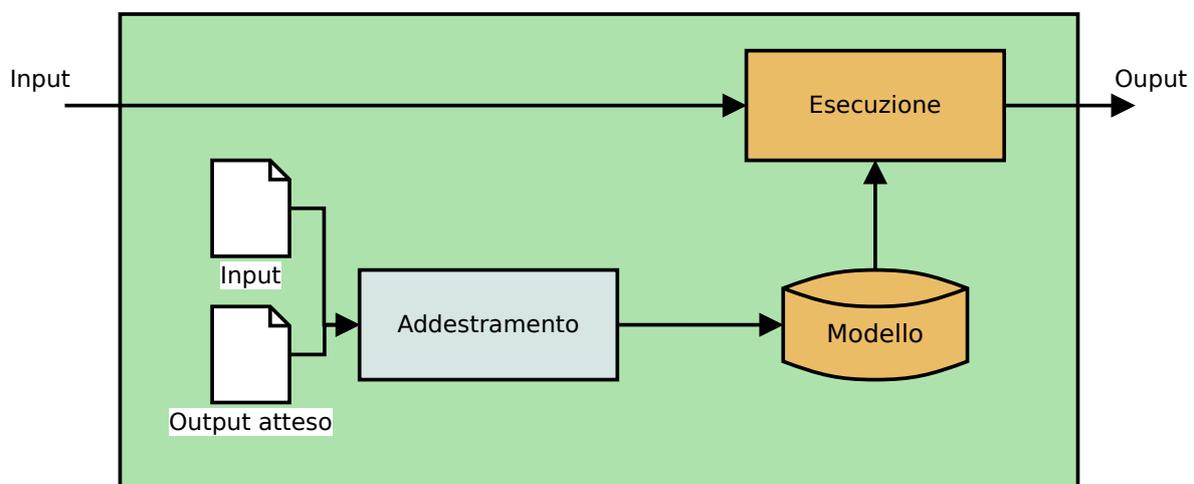


Figura 2: Generico sistema di apprendimento automatico supervisionato

Durante la fase di addestramento il sistema utilizza i dati di esempio (quindi sia l'input che l'output atteso) per trovare la funzione migliore che sia in grado di convertire un input generico nell'output atteso. Normalmente la funzione viene descritta attraverso dei parametri che vengono immagazzinati in un database (spesso realizzato attraverso semplici file) che viene solitamente chiamato modello.

Nella fase di esecuzione il modello così ricavato viene utilizzato per produrre l'output atteso, dato un input.

#### 3.2 Part-of-Speech tagging

Una delle problematiche che si riesce a trattare adeguatamente con l'apprendimento automatico supervisionato è il part-of-speech tagging, cioè l'etichettatura delle parti del discorso con etichette riferite alle classi grammaticali (nomi, verbi, aggettivi, articoli. . .). Esistono numerosi software opensource che implementano part-of-speech tagger di buona qualità, fra cui Hunpos [Mivoq(2014-2018)] e FreeLing [TALP Research Center, UPC(2008-2018a)]; questi software ricevono in input una lista di stringhe e producono in output una lista di etichette. Le etichette possono

variare sia a seconda del software utilizzato che a seconda dei dati con cui è stato effettuato l'addestramento, ma, in modo più o meno dettagliato, codificano informazioni di tipo grammaticale.

Nelle due tabelle 1 e 2 sono mostrati due esempi di etichette utilizzate rispettivamente dal software di sintesi vocale MaryTTS e (etichette basate sul formato TanL [Medialab Unipi(2009-2018)]) da FreeLing (etichette definite da FreeLing [TALP Research Center, UPC(2008-2018b)]) per il compito del pos-tagging.

Input	Output atteso	Descrizione
Chi	PRnn	Pronome relativo
conosce	Vip3s	Verbo indicativo presente 3 persona singolare
l'	RDns	Articolo determinativo singolare
apprendimento	Sms	Sostantivo maschile singolare
automatico	Ams	Aggettivo maschile singolare
?	FS	Punto

Tabella 1: Esempio di pos-tag (MaryTTS)

Input	Output atteso	Descrizione
Chi	PT0CS00	Pronome interrogativo singolare
conosce	VMIP3S0	Verbo indicativo presente 3 persona singolare
l'	DA0MS0	Articolo determinativo maschile singolare
apprendimento	NCMS000	Sostantivo maschile singolare
automatico	AQ0MS00	Aggettivo maschile singolare
?	Fit	Punto interrogativo

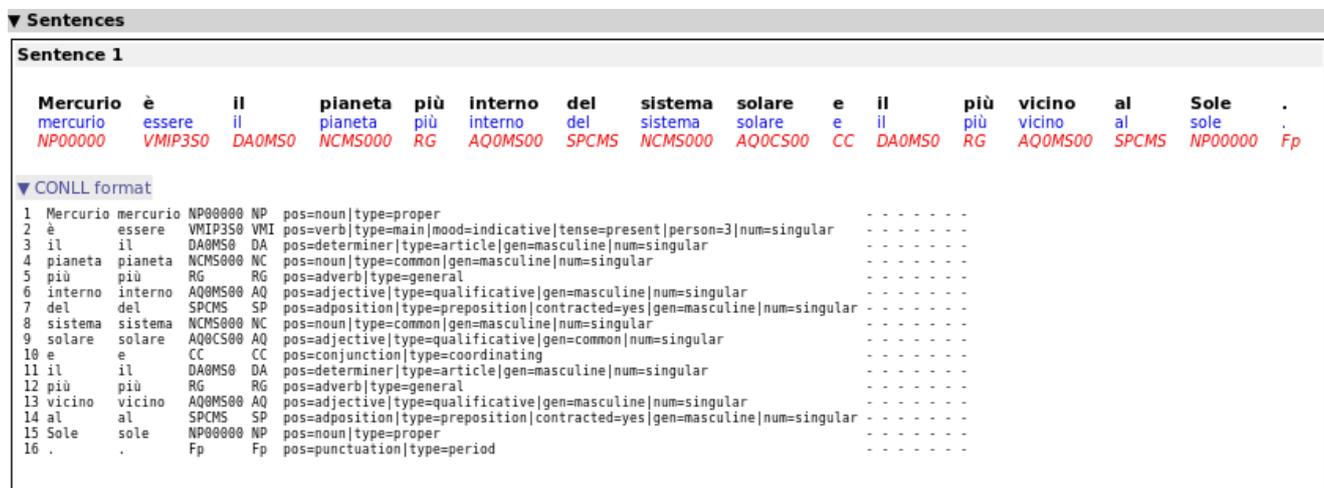
Tabella 2: Esempio di pos-tag (FreeLing)

## 4 Specifiche di progetto e riferimenti

Lo scopo del progetto è realizzare una piattaforma per raccogliere dati dagli utenti, durante il loro normale utilizzo di un servizio (preparazione e svolgimento di esercizi). Si suggerisce, per semplicità e disponibilità di software che realizzano questo compito, di concentrarsi sull'analisi grammaticale; la proponente valuterà comunque con piacere proposte alternative.

Nel caso dell'analisi grammaticale i dati da raccogliere saranno simili a quelli mostrati nelle tabelle 1 e 2. I dati dovranno essere immagazzinati opportunamente e dovranno poter essere facilmente esportati in uno o più formati leggibili dal computer (per definire i formati specifici si consiglia di prendere spunto dal formato utilizzato per il corpus Ancora [Ancora annotators(2003-2018)] o dai formati utilizzati per i corpora Universal Dependencies [Universal Dependencies contributors(2014-2018)]).

La piattaforma dovrà consentire di caricare delle frasi, per creare degli esercizi sulla base di queste. Per questa parte ci si può ispirare alla pagina dimostrativa di FreeLing [TALP Research Center, UPC(2008-2018c)], dove, al caricamento di una frase, è possibile ottenere i risultati in figura 3.



▼ Sentences

**Sentence 1**

Mercurio è il pianeta più interno del sistema solare e il più vicino al Sole .

mercurio essere il pianeta più interno del sistema solare e il più vicino al sole .

NP00000 VMIP3S0 DA0MS0 NCMS000 RG AQ0MS00 SPCMS NCMS000 AQ0CS00 CC DA0MS0 RG AQ0MS00 SPCMS NP00000 Fp

▼ CONLL format

1	Mercurio	mercurio	NP00000	NP	pos=noun type=proper	
2	è	essere	VMIP3S0	VMI	pos=verb type=main mood=indicative tense=present person=3 num=singular	
3	il	il	DA0MS0	DA	pos=determiner type=article gen=masc num=singular	
4	pianeta	pianeta	NCMS000	NC	pos=noun type=common gen=masc num=singular	
5	più	più	RG	RG	pos=adverb type=general	
6	interno	interno	AQ0MS00	AQ	pos=adjective type=qualificative gen=masc num=singular	
7	del	del	SPCMS	SP	pos=adposition type=preposition contracted=yes gen=masc num=singular	
8	sistema	sistema	NCMS000	NC	pos=noun type=common gen=masc num=singular	
9	solare	solare	AQ0CS00	AQ	pos=adjective type=qualificative gen=common num=singular	
10	e	e	CC	CC	pos=conjunction type=coordinating	
11	il	il	DA0MS0	DA	pos=determiner type=article gen=masc num=singular	
12	più	più	RG	RG	pos=adverb type=general	
13	vicino	vicino	AQ0MS00	AQ	pos=adjective type=qualificative gen=masc num=singular	
14	al	al	SPCMS	SP	pos=adposition type=preposition contracted=yes gen=masc num=singular	
15	Sole	sole	NP00000	NP	pos=noun type=proper	
16	.	.	Fp	Fp	pos=punctuation type=period	

Figura 3: Demo di FreeLing

Per la parte di preparazione degli esercizi si può creare un servizio simile in cui i risultati siano modificabili, al fine di poter applicare eventuali correzioni.

Per la parte di svolgimento si può invece pensare di preparare un'interfaccia simile in cui i campi siano però da riempire manualmente. La correzione può avvenire mediante confronto o con l'informazione salvata nel database o con le informazioni ricavate automaticamente. L'allievo che decide di affrontare un esercizio dovrebbe poter scegliere i propri insegnanti, in modo da poter privilegiare, per le correzioni, quelle basate sui dati dei propri insegnanti.

L'esempio appena descritto vuole servire da linea guida per la progettazione e chiarire almeno in parte gli obiettivi della piattaforma, ma non è obbligatorio realizzare necessariamente ogni funzionalità descritta, né è obbligatorio implementare il tema dell'analisi grammaticale. Sarà quindi possibile discutere eventuali variazioni con la proponente.

La progettazione e l'implementazione dovranno tener conto della manutenibilità e dell'estensibilità della piattaforma, cercando in particolare di permetterne il riutilizzo per altri problemi simili. Idealmente la piattaforma dovrebbe consentire la raccolta di dati in più lingue.

I commenti al codice, i nomi delle funzioni e delle variabili dovrebbero essere esplicativi ed in lingua inglese. I commenti dovrebbero essere in misura tale da permettere la comprensione del codice: un buon commento non dovrebbe descrivere la sintassi del linguaggio di programmazione in uso (quindi evitare commenti come `i = a + b; /* assign to i the sum of a and b */`), ma piuttosto descrivere lo scopo di tutti quei blocchi di codice la cui comprensione non sia già resa immediata da una corretta scelta della nomenclatura (per es.: `digit_value = digit_character - '0'; /* this line assumes that digits' characters maps to consecutive integers, ordered by value, like in ASCII character set */`).

## 5 Scelte tecnologiche

Viene lasciato un ampio margine in merito alle tecnologie impiegate: la piattaforma può essere realizzata sia come applicazione web che come applicazione mobile. Per immagazzinare i dati si suggerisce di appoggiarsi ad un servizio esistente, come Firebase [Google(2011-2018)], in modo da semplificare al massimo le problematiche di installazione, ma possono essere utilizzati anche altri sistemi. Si suggerisce l'utilizzo di un software opensource per pos-tagging, come Hunpos [Mivoq(2014-2018)] o FreeLing [TALP Research Center, UPC(2008-2018a)], ma, previa accordo con la proponente, possono essere valutati altri strumenti e compiti diversi rispetto al pos-tagging.

## A Note sul capitolato d'appalto

Il presente capitolato d'appalto rappresenta una prima bozza di lavoro per aiutare il fornitore a capire le tematiche del progetto e del tipo di sviluppo che sarà necessario. La proponente si rende disponibile per ogni ulteriore chiarimento.

### A.1 Aspettative della proponente

La proponente si aspetta di veder realizzata una piattaforma che possa essere immediatamente utilizzata per agevolare la raccolta di dati che possano essere impiegati per la realizzazione di servizi utili basati sull'apprendimento automatico. Per quanto consigliato, in virtù della semplicità realizzativa, il tema dell'analisi grammaticale non è vincolante.

La piattaforma realizzata, salvo ulteriori accordi, resterà di proprietà esclusiva del fornitore. La proponente incoraggia, ove possibile, il rilascio con licenze opensource e preferibilmente con una licenza in stile BSD/MIT.

### A.2 Contatti

La proponente potrà essere contattata in qualsiasi momento attraverso l'indirizzo email `tech@mivoq.it`, al quale risponde il reparto tecnologico dell'azienda. Le email dovranno contenere in oggetto la sigla “[UNIPD-SWE]” e dovranno essere indirizzate all'attenzione di Giulio Paci, che sarà il referente principale per il progetto.

In alternativa è possibile telefonare al numero 0490998335, al quale risponde il reparto tecnologico dell'azienda. In questo caso sarà sufficiente specificare che si chiama a proposito del progetto di “Ingegneria del software”.

## B Note sulla proponente

Mivoq è un'azienda, nata nel 2013, che si occupa di sintesi vocale, con lo scopo di permettere ad ogni singolo utente di avere la propria voce digitale che possa rappresentarlo anche dove normalmente non è presente, attraverso applicazioni innovative per la lettura di SMS con la voce del mittente, la lettura di post su Facebook con la voce dell'autore, la personalizzazione di assistenti virtuali con la voce dell'utente o la produzione della propria voce quando questa è venuta meno a causa di un'operazione o di una malattia (si pensi per esempio al fisico Stephen Hawking).

L'obiettivo di Mivoq non è di realizzare queste applicazioni, ma di fornire la tecnologia necessaria a fare in modo che vengano realizzate. Con questo progetto intendiamo migliorare il processo di sviluppo di questa tecnologia introducendo un'importante strumento di debug.

## Riferimenti bibliografici

- [Mivoq(2014-2018)] Mivoq. (2014-2018) Hunpos. [Online]. Available: <https://github.com/mivoq/hunpos>
- [TALP Research Center, UPC(2008-2018a)] TALP Research Center, UPC. (2008-2018) Freeling home page. [Online]. Available: <http://nlp.lsi.upc.edu/freeling/node/12>
- [HT Tecnologie Umanistiche(2012-2018)] HT Tecnologie Umanistiche. (2012-2018) Analisi grammaticale 2018 di archivium. [Online]. Available: <https://www.analisi-grammaticale.biz/>
- [VoxForge(2006-2018)] VoxForge. (2006-2018) Voxforge. [Online]. Available: <http://www.voxforge.org/>
- [Google(2006-2018)] Google. (2006-2018) Translate. [Online]. Available: <https://translate.google.com/>
- [Medialab Unipi(2009-2018)] Medialab Unipi. (2009-2018) Tanl pos tagset. [Online]. Available: [http://medialab.di.unipi.it/wiki/Tanl\\_POS\\_Tagset](http://medialab.di.unipi.it/wiki/Tanl_POS_Tagset)
- [TALP Research Center, UPC(2008-2018b)] TALP Research Center, UPC. (2008-2018) Freeling 4.0 user manual - italian. [Online]. Available: <https://talp-upc.gitbook.io/freeling-4-0-user-manual/tagsets/tagset-it>
- [Ancora annotators(2003-2018)] Ancora annotators. (2003-2018) Ancora corpus. [Online]. Available: <http://clic.ub.edu/corpus/en>
- [Universal Dependencies contributors(2014-2018)] Universal Dependencies contributors. (2014-2018) Universal dependencies. [Online]. Available: <http://universaldependencies.org/>
- [TALP Research Center, UPC(2008-2018c)] TALP Research Center, UPC. (2008-2018) Freeling 4.1 - demonstration. [Online]. Available: <http://nlp.lsi.upc.edu/freeling/demo/demo.php>
- [Google(2011-2018)] Google. (2011-2018) Firebase. [Online]. Available: <https://firebase.google.com/>