Università degli studi di Milano-Bicocca

Data Science Lab in Public Policies and Services

Modulo Big Data in Public and Social Services

Sistema di auto-completamento semantico di skills

Simone D'Amico – 850369

Tutor: Andrea Seveso

AGENDA

- Contesto e obiettivi del progetto.
- Related works e dati a disposizione.
- Definizione formale del problema.
- Implementazione del sistema.
- User test.
- Futuri sviluppi e conclusioni.

CONTESTO

- Negli ultimi anni si è assistito alla nascita di nuove professioni che coinvolgono e richiedono un numero sempre maggiore di conoscenze e competenze.

- Diventa importante individuare correttamente le competenze necessarie per svolgere un determinato lavoro.

- Aree di studio come il *Labor Market Intelligence* forniscono utili strumenti per supportare le attività decisionali riguardo il mercato del lavoro
  - monitoraggio e analisi di annunci di lavoro online *(OJV)*

- Tassonomia standard europea delle occupazioni e delle competenze (E.S.C.O.):

> *«La classificazione E.S.C.O. identifica e classifica skills, competenze, qualifiche e occupazioni rilevanti per il mercato del lavoro europeo, l'istruzione e la formazione»*
>
> *- Handbook, E. S. C. O. European Skills, Competences, Qualifications andOccupations (2017). EC Directorate E.*

OBIETTIVI DEL PROGETTO

- Sviluppare di un sistema di suggerimento delle skills definite dalla tassonomia E.S.C.O.

- In particolare lo studio proposto cerca di affrontare due domande di ricerca:

  Q1) *Quali strumenti possono essere utilizzati per produrre un sistema che possa suggerire parole che siano, non solo sintatticamente ma anche semanticamente simili?*

  Q2) *In che modo tale sistema possa calcolare la similarità tra le parole e suggerire quelle più adatte ad un particolare contesto?*

RELATED WORKS

L'idea alla base di questo sistema nasce da altri lavori:

- *NEO: A Tool for Taxonomy Enrichment with New Emerging Occupations*[1]
  I. Impari i word embeddings di concetti e entità delle tassonomia preservando le relazioni tassonomiche.
  II. Suggerisca nuove entità per E.S.C.O. estratte da un corpus testuale .
  III. Valuti, tramite apposite misure, la loro idoneità come entità di differenti concetti tassonomici.

- *Skills4job:* un lavoro ancora in corso, in cui si richiede all'utente l'inserimento di skills di interesse e si fornisce il nome di un lavoro ad esse correlato.

---

[1] *Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Seveso,A. (2020). NEO: A Tool for Taxonomy Enrichment with New Emerging Occupations.*

BASE LINE

- Il sistema utilizza la *similarità del coseno* per calcolare:
  - La similarità tra l'input scritto dall'utente e le skills di E.S.C.O.
  - La similarità tra le skills già inserite e le altre non ancora inserite

- Aggrega queste due similarità in un'unica misura suggerendo poi le skills con il valore più altro per questa similarità

- Dati a disposizione:
  - L'insieme *S* delle skills E.S.C.O., in totale 2319.
  - Gli embeddings imparati tramite FastText: 113656 vettori
    - Uso di FastText permette di ottenere i vettori non solo delle skills, ma anche delle loro sotto-parole

DEFINIZIONE DEL PROBLEMA – CALCOLO SIMILARITÀ

- Un'ulteriore importante concetto è quello di *Context:*
  - L'insieme delle skills già inserite dall'utente

- Per la similarità tra l'input i scritto dall'utente e le skills si calcola:

$$A = \{(s, sim(i, s)) \mid s \in S \ \& \ s \notin C\}$$

- Per la similarità tra le skills già inserite e le altre non ancora inserite si calcola:

$$B = \{(s, sim_{avg}(s, C)) \mid s \in S \ \& \ s \notin C\}$$

- Dove $sim_{avg}(s, C)$ è definita come la media delle similarità della skill *s* con le skills nel *context C:*

$$sim_{avg}(s, C) = \frac{\sum_{c \in C} sim(s, c)}{|C|}$$

DEFINIZIONE DEL PROBLEMA –
MISURA AGGREGATA

- Una volta calcolati gli insiemi A e B e le relative similarità, si calcola la similarità aggregata:

$$sim_s = \alpha * sim_A + (1 - \alpha) * sim_B$$

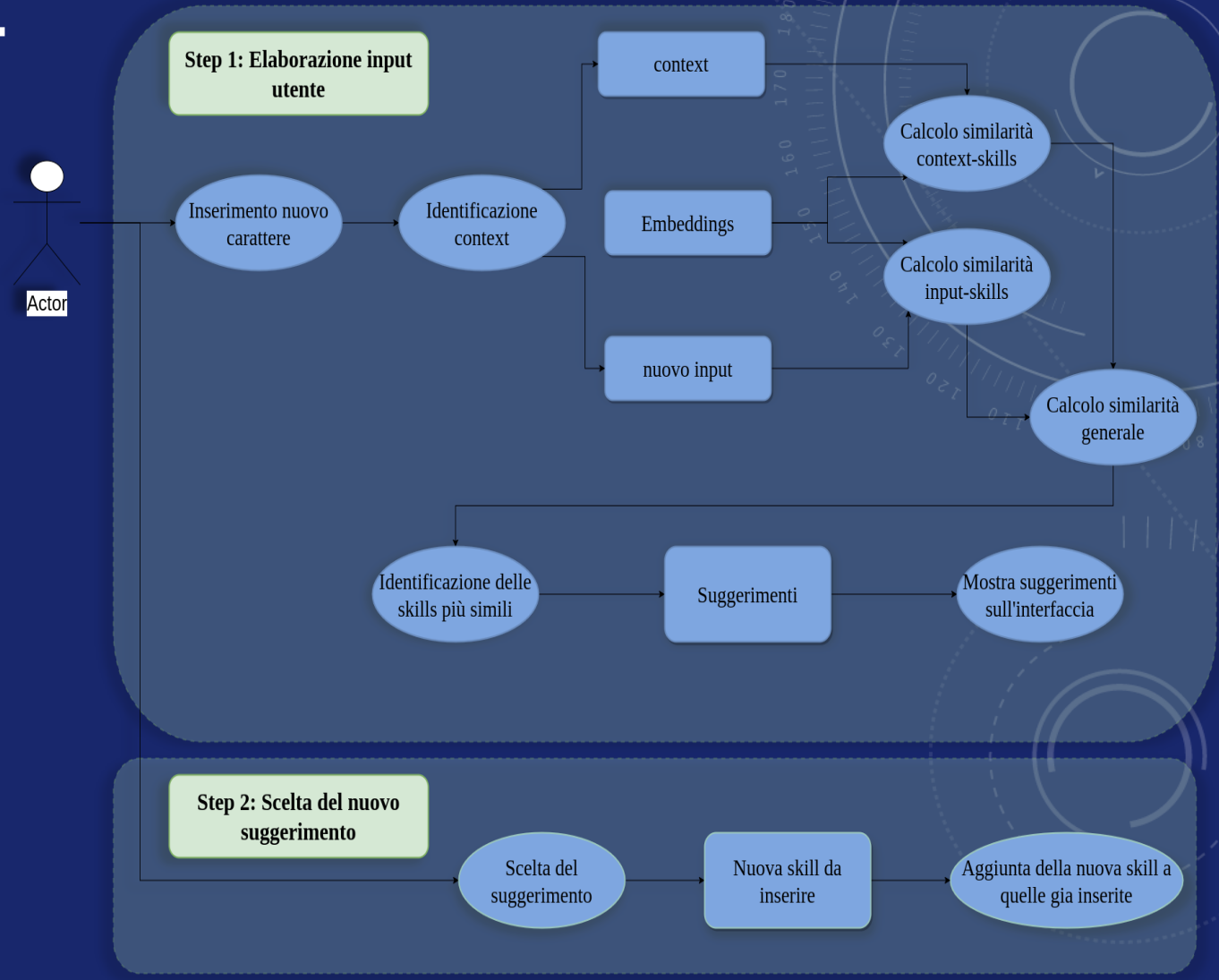
$$\forall s \in S \,\&\, s \notin C$$
$$(s, sim_A \in A)$$
$$(s, sim_B \in B)$$

- Per ogni possibile skill da suggerire si ottiene la sua similarità sia per un nuovo input dell'utente e sia per le skill già inserite

IMPLEMENTAZIONE

- Il sistema è stato sviluppato in *Python* con l'utilizzo di *Jupyter Notebook*

- L'architettura è formato da due componenti:
  - AutoCompleteManager: gestisce l'interazione con l'utente, il calcolo delle similarità e propone i migliori suggerimenti all'utente.
  - FormUI: definisce l'interfaccia utente realizzata tramite i *widgets* di *Jupyter Notebook*



MODULO AUTOCOMPLETEMANAGER – CALCOLO DELLE SIMILARITÀ

Algorithm 1: *get_best_similarity_skill:* calcola le similarità secondo l'eq. (5)

Input: α, *new_input: l'input dell'utente, context: lista delle skills già inserite, num_suggests: numero di suggerimenti da mostrare, skills_list: le skills E.S.C.O.*

Output: Le skill più simili sia al contest (se presente) che all'input con i relativi valori di similarità

```
1  sim_A ← get_skills_input_similarity(new_input, context)
2  if context == [] then
3      most_similar ← sorted(sim_A)[0 : num_suggests]
4      return most_similar
5  sim_B ← get_skills_context_similarity(context)
6  res ← dict()
7  for s in sim_A.keys() do
8      res[s] = α * sim_A[s] + (1 − α) * sim_B[s]
9  most_similar ← sorted(res)[0 : num_suggests]
10 return most_similar
```

Calcolo delle similarità considerando solo il nuovo input

Calcolo delle similarità considerando solo *context*

Similarità aggregata

MODULO AUTOCOMPLETEMANAGER – LISTENER DELL'AREA DI TESTO

All'area di testo è associata una funzione listener che rimane in esecuzione aspettando l'input dell'utente, calcola le similarità e crea i bottoni per i migliori suggerimenti

Algorithm 2: *suggests_manager:* calcola le similarità ad ogni nuovo input e mostra i suggerimenti più adatti

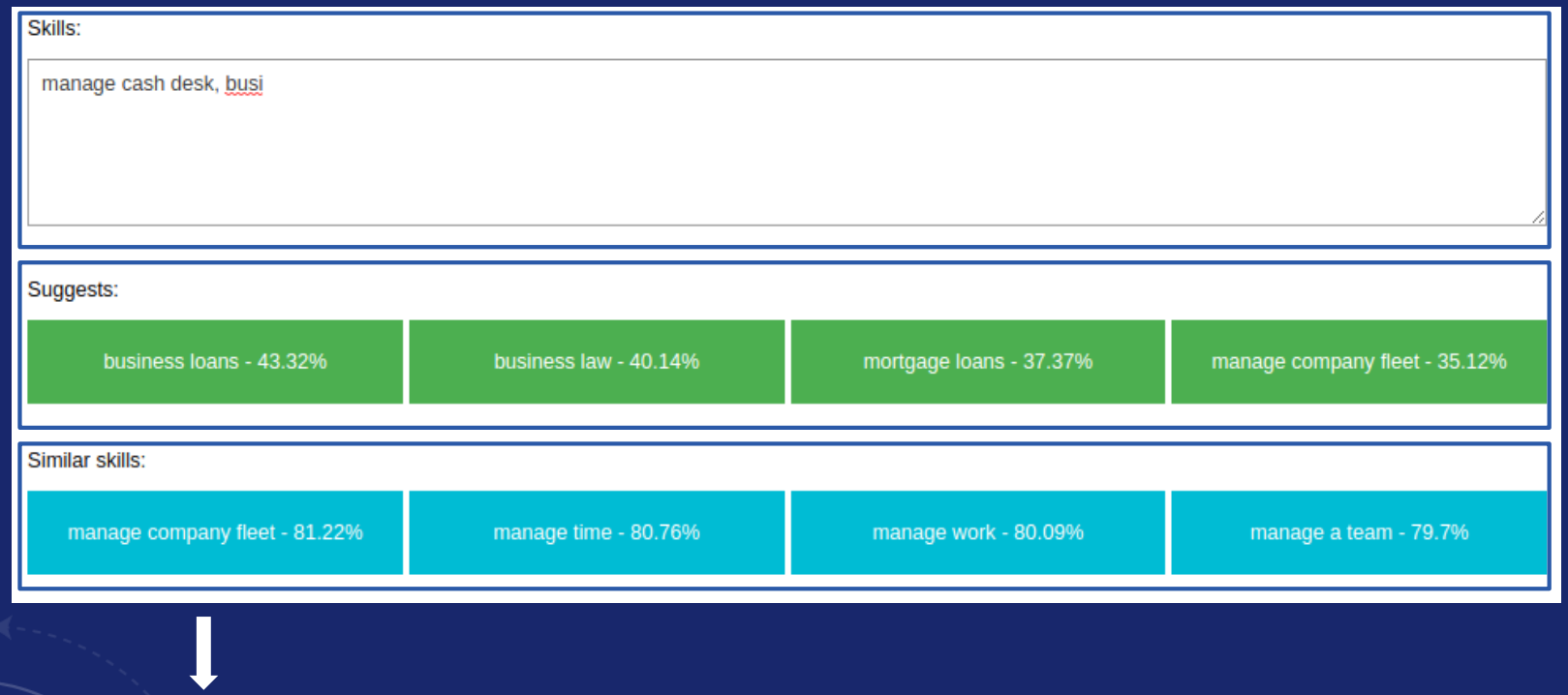
Input: *num_suggests*, *lista_unique*, *primary_key*
Output: Le proprietà definite con le API generiche

```
1 while True do
2     new_input ← strings inserita fino a quel momento
3     context ← lista delle skills già inserite
4     suggests ← get_best_similarity_skill(new_input, context)
5     for s in suggests do
6         Crea e visualizza il bottone con descrizione s e la sua
          similarità con new_input
```

Calcolo della similarità aggregata

MODULO FORMUI

- Il modulo crea il layout dell'interfaccia e permette di definire i bottoni associati ai vari suggerimenti mostrati
- L'interfaccia è divisa in tre sezioni:



Una text area dove inserire le skills di interesse e che viene aggiornata aggiungendo la nuova skill selezionata

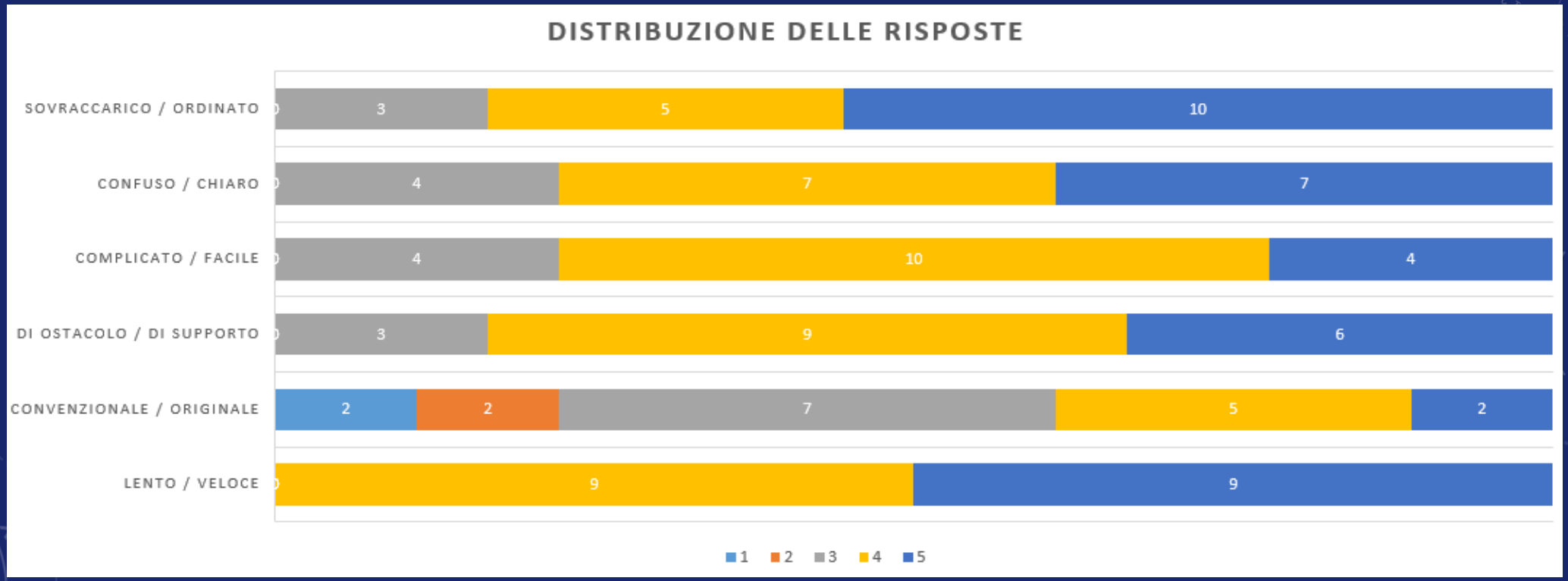
L'area *Suggests* mostra le skills più simili considerando sia il *context* che il nuovo input

L'area *Similar skills* mostra le skills più simili considerando solo il *context*

USER TEST

- Gli utenti hanno utilizzato il sistema per almeno cinque minuti

- Poi hanno risposto ad un questionario di valutazione di alcune caratteristiche su una scala da 1 a 5



SVILUPPI FUTURI

- Considerare anche la gerarchia a più livelli della classificazione E.S.C.O. e utilizzare sia le similarità che una misura di distanza delle skills
  - Miglioramento del valore di similarità complessivo
- Filtrare determinati annunci con qualche criterio (anno, area geografica, …) e calcolare solo i vettori embeddings di questi annunci
  - Eliminazione di annunci di scarso interesse per lo specifico scenario di utilizzo
- Utilizzare diverse misure si similarità oltre a quella del coseno

CONCLUSIONI

- (Q1) *Quali strumenti possono essere utilizzati per produrre un sistema che possa suggerire parole che siano, non solo sintatticamente ma anche semanticamente simili?*
  - Modello di word embeddings FastText con cui calcolare le similarità delle varie skills.

- (Q2) *In che modo tale sistema possa calcolare la similarità tra le parole e suggerire quelle più adatte ad un particolare contesto?*
  - Similarità del coseno.
  - Metrica aggregata per il calcolo delle similarità.

- Il meccanismo di suggerimento è indipendente dallo scenario, funziona non solo per skills tecniche.

- Logica separata dall'interfaccia.
  - inseribile all'interno di applicazioni web o di interfacce di programmi desktop.

REFERENCES

- Giabelli, A., Malandri, L., Mercorio, F., & Mezzanzanica, M. (2020). *GraphLMI: A data driven system for exploring labor market information through graph databases.* Multimedia Tools and Applications, 1-30.

- Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Seveso,A. (2020). *NEO: A Tool for Taxonomy Enrichment with New Emerging Occupations.*

- CEDEFOP: Real-time labour market information on skill require-ments: Setting up the eu system for online vacancy analysis. https://goo.gl/5FZS3E (2016).

- Handbook, E. S. C. O. European Skills, Competences, Qualifications and Occupations (2017). EC Directorate E.

GRAZIE PER L'ATTENZIONE

Un ringraziamento particolare ad Andrea per il suo prezioso aiuto