Università degli Studi di Milano
Facoltà di Scienze Matematiche Fisiche e Naturali
Corso di Laurea in Informatica per la Comunicazione

PROFILE MATCHING IN ONLINE SOCIAL NETWORKS

Relatore: Prof. Sabrina GAITO

Correlatore: Dr. Matteo Zignani

Tesi di:
Mattia Dimauro
Matricola: 808902

Anno Accademico 2013-2014

dedicato a ...

Prefazione

Prefazione lalala.

Organizzazione della tesi

La tesi è organizzata come segue:

- nel Capitolo 1 ....

Ringraziamenti

Ringrazio, grazie.

Indice

Capitolo 1

Introduzione

Ciao volevo bla bla lba

Capitolo 2

Pattern comportamentali e costruzione di features

Gli individui mostrano spesso pattern comportamentali nella scelta dei loro usernames. Questi pattern, risultanti in ridondanza di informazione, possono essere utili per identificare individui su diversi social networks. I soggetti potrebbero evitare queste ridondanze selezionando usernames in modo che risultino completamente diversi dai loro altri usernames. In questa maniera gli usernames risulterebbero essere talmente differenti che dato uno username, nessuna informazione riguardante gli altri usernames potrebbe essere estratta. Idealmente per raggiungere questo stato di indipendenza tra usernames, l'individuo dovrebbe scegliere uno username che presenti entropia massima. Ovvero uno username composto da una lunga sequenza di caratteri, lunga quanto il massimo consentito dal sistema, senza ridondanze: una sequenza di caratteri completamente casuale. Sfortunatamente, tutti questi requisiti non vengono incontro alle abilità umane. Gli esseri umani hanno difficoltà a memorizzare lunghe sequenze, con la possibilità della memoria a breve termine di ricordare 7 ± 2 elementi. Queste limitazioni risultano condurre gli individui a selezionare generlamente usernames *non lunghi*, *non casuali* e che presentano *abbondante ridondanza*. Queste proprietà possono essere catturate adottando features specifiche.

Possiamo suddividere questi pattern comportamentali in tre categorie:

1. Pattern dovuti a limitazioni umane
2. Fattori esogeni
3. Fattori endogeni

Discuteremo dei comportamenti di ognuna di queste categorie elencate e delle features che possono essere estrapolate sfruttando questi pattern.

2.1 Pattern dovuti a limitazioni umane

definizione notazione Ci riferiremo a Individuo, username, prior usernames, candidate username.

Username identici Uno studio condotto da Zafarani dimostra che il 59% degli individui preferisce usare lo stesso username reiteratamente, principalmente per facilitá nel ricordarlo.[citazione]

Quindi se un candidate username c compare tra i prior usernames U, vi é una forte indicazione che che potrebbe essere associato allo stesso individuo a cui sono associati i prior usernames. Considereremo dunque di utilizzare il numero di candidate username presenti tra i prior usernames come feature.

Username Length Likelihood Allo stesso modo, gli utenti hanno tipicamente un insieme di potenziali usernames dal quale ne estraggono uno quando richiesto di crearne uno nuovo. Questi usernames hanno differenti lunghezze e dunque é possibile calcolarne una distribuzione. Consideriamo l_c la lunghezza del candidate username e l_u la lunghezza di uno username $u \in U$. Assumiamo che per ogni nuovo username deciso di creare é probabile il verificarsi di

$$\min l_u \leq l_c \leq \max l_u$$

Ad esempio se un individuo é solito a scegliere username di lunghezza 8 o 9 caratteri, é improbabile che questo consideri di crearne uno con lunghezze minori o maggiori. Considereremo quindi le lunghezza del candidate username e la distribuzione delle lunghezze dei prior usernames come features. La distribuzione verrá rappresentata da un numero fisso di features, descrivendola come

$$(\mathbb{E}[l_u], \sigma[l_u], \text{med}[l_u], \min l_u, \max l_u)$$

Unique username creation Likelihood

Limited Vocabulary

Limited Alphabet

2.2 Fattori esogeni

2.3 Fattori endogeni

Capitolo 3

Collezione dataset

Esistono più modalità attraverso le quali è possibile collezionare dati riguardanti stessi individui presenti su diversi social networks. Un modo semplice consisterebbe nel organizzare un questionario dove si richiede agli utenti di elencare i propri profili. Questo metodo permette di raccogliere una quantità di dati spesso limitata. Esistono compagnie, alcune risultano essere gli stessi social networks, che richiedono queste informazioni ai propri utenti ma queste non sono disponibili pubblicamente. Fortunatamente esistono servizi di *social network aggregation* che permettono di collezionare contenuti da diversi *social network services* in una presentazione unificata. Il compito svolto da un *social network aggregator*, che raggruppa insieme informazioni in un singolo luogo, supporta i propri utenti a riunire molteplici profili di social network in un singolo profilo e aiuta a tenere traccia delle attività che avvengono sui diversi profili semplificando la *social networking experience* dell'utente. La maggior parte di questi servizi non permette l'accesso pubblico alle informazioni aggregate, sono per lo più uno strumento per l'utilizzatore per seguire i propri profili, permettendo di avere tutte le notifiche relative in unico luogo, o di pubblicare lo stesso contenuto in più profili in una volta sola.Questa tipologia di aggregatori non sono frutto di interesse per la collezione di informazioni che cerchiamo. Prenderemo in considerazione invece quegli aggregatori che permettono di condividere queste informazioni aggregate con altre persone. Attraverso questi è possibile visualizzare tutte le attività di un utente sui diversi social network che questo a deciso di far seguire all'aggregatore. Ovviamente è possibile, ed è l'informazione che andremo a estrarre, risalire su quale social network è stata eseguita tale azione e avere cosi un elenco di profili di social network services appartenenti alla stessa persona.

3.1 Scraping Alternion

Alternion - All your social web and email in one place - è un aggregatore i cui servizi e features lo fanno rientrare tra la seconda classe di *social network aggregator* descritta. Attraverso *Alternion* è in fatti possibile per un utente condividere con altri contatti le informazioni riguardanti i propri profili di diversi online social networks. Il numero di utenti iscritti al servizio non è noto. Alternion dichiara di permettere ai propri utenti di aggregare un grosso numero di online social networks, dai più popolari come *Facebook*, *Twitter*, *Google+*, *LinkedIn*, *Flickr* fino a più di 220 online social networks. Di seguito spiegheremo l'approccio utilizzato al fine di ottenere i profili degli utenti iscritti al servizio e di recuperare da questi le informazioni riguardati i profili dei loro social networks. Alternion non dispone di un servizio per recuperare i dati attraverso un'interfaccia *web (API)*, con richieste e risposte documentate. Dovremo dunque procedere analizzando le pagine web restituite svolgendo un'attività conosciuta come *web scraping*. Il *web scraping* o *web data extraction* è una tecnica di estrazioni di dati da pagine web. L'estrazione di dati viene automatizzata attraverso un software che simula un utente umano nell'esplorazione di documenti presenti sulla rete internet. Il software deve quindi implementare il protocollo HTTP, fondamento per la comunicazione di dati per il World Wide Web per recuperare il documento attraverso la rete internet. Questi documenti, tipicamente descritti attraverso un linguaggio di markup, sono pensati per rappresentare un'interfaccia grafica per l'utente. Ottenuto il documento il *web scraper* si occupa di analizzarne i dati strutturati ricercandone quelli di interesse. La pagina non viene interpretata visivamente ma esaminandone il contenuto descritto con il linguaggio di markup, tipicamente HTML. Questa ricerca può essere effettuata con diversi approcci: dal più semplice, seppur potente, *text grepping* combinato con *regular expression matching* o può prevedere un'analisi più strutturata della pagina attraverso una tecnica di *DOM parsing*. Per i nostri scopi utilizzeremo entrambe queste tecniche. Esistono altre tecniche e metodologie per eseguire *web scraping*, ma i dettagli esulano dallo scopo di questa tesi.

3.1.1 Ottenere i profili

Siamo interessanti a ottenere un considerevole numero di profili di utenti che utilizzano l'aggregatore in analisi. Alternion non prevede una funzionalità per mostrare l'elenco completo dei profili iscritti al proprio servizio. Permette invece di reperirne un sotto insieme di questo attraverso una funzionalità di ricerca, divisibile in due classi: la ricerca parametrizzabile secondo alcuni criteri o la presentazione casuale di profili. La prima permette di interrogare il sistema per estrarne i profili corrispondenti ad alcuni parametri di ricerca come Nome, Sesso, Età, Paese di provenienza, Educazione,

Interessi, etc La seconda funzionalità consente di ottenere ad ogni richiesta un profilo selezionato casualmente.[1] Ho deciso di accantonare questa seconda via per due ragioni. La selezione casuale potrebbe potenzialmente portare a richiedere più volte lo stesso profilo, dipendentemente dalla bontà (che non è stata testata) della casualità con cui il profilo viene estratto. Vogliamo inoltre limitare l'estrazione di profili non appartenenti a persone fisiche: alcuni profili presenti fanno infatti riferimento a prodotti o società e aziende. Concludo di optare per la ricerca parametrizzata, in particolare, la ricerca per nome. Come lista di parametri per l'interrogazione del sistema useremo una lista di nomi propri [2], formata da 4275 nomi femminili e 1219 maschili. Utilizzando strumenti messi a disposizione da Google[3] per analizzarne i pacchetti HTTP scambiati tra il server di Alternion e il client Google Chrome, viene identificata la richiesta HTTP da eseguire per interrogare il server per farsi restituire la pagina web contenente la lista di utenti corrispondenti al parametro di ricerca. Eseguiremo quindi una richiesta per ogni nome presente nelle nostre liste di nomi. Da questa, tramite tecniche di scraping descritte precedentemente, estrapoliamo per ogni utente l'*URL* idendificativo della risorsa. Una volta collezionati tutti gli URL, dove ogni indirizzo corrisponde a un utente di Alternion, potremo recuperare la pagina profilo di questi utenti e cercare al suo interno le informazioni riguardanti i loro social networks. Per la persistenza dei dati useremo MongoDB, un database NoSQL document-oriented, che utilizza JSON come data model.

3.1.2 Profili recuperati

Sono stati recuperati 15341 profili di Alternion, di cui 11274 presentano almeno due profili di social networks. Il numero di profili non è ingente, in quanto la funzionalità ricerca permette di recuperare un massimo di 30 profili per richiesta, ad esempio solo trenta profili delle persone che si chiamano 'Anna'. Ad ogni modo si è notato che è possibile scalare, se non verticalmente per numero di users, orizzontalmente per numero di profili per users. Ogni utente ha in media ∼4,6 OSNS collegati, distribuiti tra un totale di 168 online social network services diversi presenti. Possiamo dunque espandere il numero di coppie di usernames secondo una combinazione semplice di n elementi di classe k, dove $n = 2$ (coppie) e k il numero di classi di OSN distinti. Ad esempio, un profilo P ha aggreagato 3 OSNs {Facebook, Twitter, Instagram } e presenta quindi uno username u per ogni profilo $U = \{u_{\text{Alternion}}, u_{\text{Facebook}}, u_{\text{Twitter}}, u_{\text{Instagram}}\}$.

[1] Più probabilmente, pseudo-casualmente!
[2] Lista reperita da census.gov
[3] Google Chrome DevTools

I sottoinsiemi di cardinalità 2 dell'insieme U sono:

- $\{u_{AL}, u_{FB}\}$
- $\{u_{AL}, u_{TW}\}$
- $\{u_{AL}, u_{INM}\}$
- $\{u_{FB}, u_{TW}\}$
- $\{u_{FB}, u_{INM}\}$
- $\{u_{TW}, u_{INM}\}$

Potenzialmente, il numero di classi di coppie di social network possibile é dimostrato essere uguale al coefficiente binomiale

$$\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k(k-1)\dots 1}$$

che puó essere scritto usando il fattoriale come

$$\frac{n!}{k!(n-k)!}$$

dove $k = 2$ (coppie) e $n = 168$, quindi 14028 coppie di social network distinte. Solamente 5855 di queste peró presentano almeno una coppia di username al suo interno. Concludendo, applicando la combinazione a 2 elementi per ogni profilo nel nostro dataset, otteniamo $\sim$170000 coppie di usernames. Con una media di $\sim$29 coppie di username per classe.

OSNS presenti {100zakladokru, 43 Things, 500px, ActiveRain, All Consuming, Alternion, Amazon, Ameba, Aminus3, Answerbag, Aol Answers, AudioBoo, Bambuser, Bebo, Blipfm, Blipfoto, Bliptv, Blog Talk Radio, Blogger, Blogmarks, Blogru, Blogs@MailRu, Bordom, BuzzFeed, Buzznet, CafeMom, CiteULike, Connotea, Current, DailyStrength, Dailymotion, Delicious, DeviantART, Diigo, Disqus, Docstoc, Douban, Dreamwidth, Dribbble, EmpoweHER, Etsy, Eventbrite, FFFFound!, Facebook, Fancy, Flickr, FoodFeed, Formspring, Fotolog, Foursquare, FunnyOrDie, GamerDNA, Gamespot, Gather, GitHub, Gizmodo, Goodreads, Google Reader, Google+, Habrahabr, Hatena Bookmark, Hatena Diary, Hatena Haiku, HubPages, Hyves, Identica, Imgly, Instagram, Instructables, IntenseDebate, Ipernity, Issuu, Jalbum, JamBase, Judy"s Book, Lafango, Lastfm, LibraryThing, LinkedIn, Listal, Lookbook-nu, Magma, MeasuredUp, Memoriru, Meneame, MetaFilter, Metacafe, Mister Wong,

Moblog, MobyPicture, Multiply, Netlog, News2ru, Newsvine, NowPublic, Pandora, Panoramio, Photobucket, Photocase, Photosightru, Picasa, Pikchur, Pinboard, Pinterest, Plancast, Plixi, Plurk, Polyvore, Posterous, Qik, Qype, RPodru, Raptr, RedBubble, RedGage, Reddit, Rooftop Comedy, SAPO Fotos, SAPO Videos, Server Fault, Six Groups,Skyrock, SlideShare, SmugMug, Soupio, SparkPeople, Squidoo, Stack Overflow, StumbleUpon, Super User, Tabulas, Technorati, ThisNext, Threadless, TravelPod, Trilulilu, Tripit, Trulia, Tumblr, Tvigleru, Twitgoo, Twitpic, Twitrpix, Twitter, UserVoice, Viddler, VideoQip, Vimeo, Wattv, We Heart It, WordPress, Worth1000, Xanga, Yahoo! Answers, YouTube, Zazzle, Zenfolio, Zillow, Zorpia, aNobii, authorSTREAM, Facebook, gdgt, I use this, iPadio, iReport, Visualizeus, wePapers}

Lo spettro di social networks presenti é ampio e variegato per dominio d'interesse, contenuti, tipologia di utenti e servizi offerti. Non é semplice crearne una tassonomia, che in ogni caso, non delineerebbe dei gruppi chiaramente separati e privi di sovrapposizioni. Notiamo la presenza dei SN piú popolari come Facebook, Twitter, Google+, passando per sistemi conosciuti per le loro funzionalitá di *Location Based Services* (LBS) come Foursquare, o sistemi distinti per contenuti a tema musicale, fotografico o video come Lastfm, Pandora, Flickr, Instagram, 500px, Youtube, Vimeo, e ancora, sistemi dedicati al blogging come Tumblr, o sistemi di social bookmarking come StumbleUpon e altri.

3.1.3 Distribuzione della similaritá degli usernames

Siamo interessati a confrontare coppie di usernames, per tentare di ricollegarle alla stessa entitá. Il caso piú semplice che potrebbe presentarsi[4] é la corrispondenza esatta tra i due usernames. Indagheremo quindi questa proprietá sul nostro dataset acquisito. Oltre a verificarne la corrispondenza esatta, ovvero che le due stringhe che formano gli username presentino gli stessi caratteri nello stesso ordine, faremo uso di alcune metriche di similaritá su stringhe. Queste sono tecniche per quantificare la dissimilaritá tra due stringhe. In particolare utilizzeremo la *distanza di Levensthein*, una metrica per misurare la differenza tra due sequenze, e *l'indice di Jaccard*, un indice statistico utilizzato per comparare la similaritá e la diversitá di insiemi campionari.

Distanze Riportiamo le statistiche su le distanze calcolate su le classi aventi almeno 1000 coppie di username (29 classi, con un totale di 56687 coppie di usernames) e su i dati globali.

[4]se non si considerano i casi di omonimia

	Mean Levensthein distance	Mean Jaccard Index
Popular classes	6.2295	0.3277
All dataset	5.8159	0.3685

Match Esatti Qui riportiamo le statistiche su i match esatti, dove le distanze di Levensthein e l'indice di Jaccard assumomo il valore 0.

	Levensthein distance = 0	Jaccard Index = 0
Popular classes	31.79%	34.03%
All dataset	27.80%	29.16%

Riportiamo alcuni grafi per illustrare alcune proprietå dei dati raccolti. Tutti i grafi qui riprodotti sono stati generati con l'ausilio di matplotlib[5], una popolare libreria open source per rappresentare set di dati con grafi bidimensionali come plots, istogrammi e bar charts, descrivibili attraverso poche linee di codice, in specifico, Python.

I grafi in Figura 1 e 2 sono degli istogrammi che rappresentano la distribuzione delle distanze di Levenshtein e dell'indice di Jaccard rispettivamente tra le coppie di username del dataset. L'istogramma in Figura 1 presenta sull'asse x le distanze e sull'asse delle y il numero di coppie che ricade in quella classe. Il numero di classi e il suo spettro riflette le distanze riscontrate per ogni coppia: da 0, distanza minima oltre la quale non é possibile scendere a 45 massima distanza presente tra due username in tutto il dataset. Per l'istogramma in Figura 2, il numero di classi o *bin* é fisso, con un range 0-1, ovvero lo stesso spettro di valori assumibili dall'indice di Jaccard. Notiamo in entrambi i grafi di Fig. 1 e 2 una distribuzione bimodale, é abbastanza semplice infatti individuare due picchi nei due istogrammi. Il primo picco, sullo 0, é facilmente riconducibile alle coppie di username identici tra di loro.

[5]http://matplotlib.org/

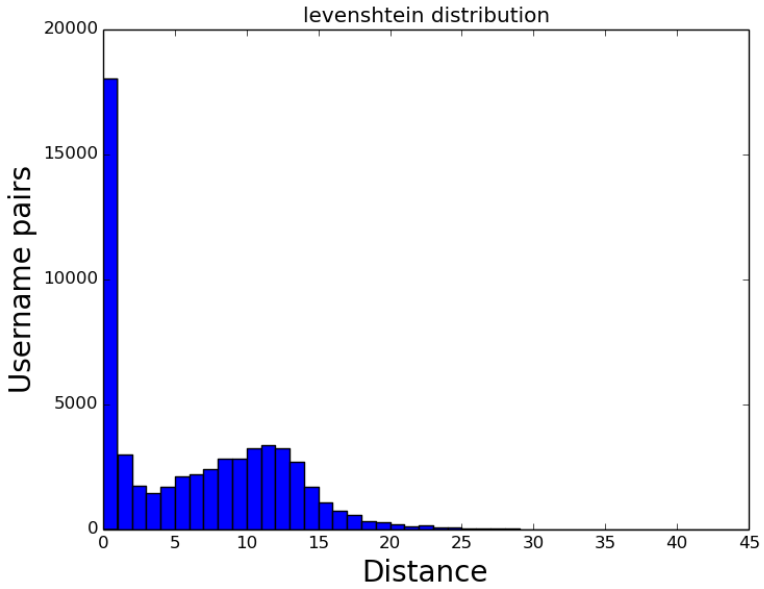


Figura 1: Distribuzione distanza di Levensthein tra coppie di usernames

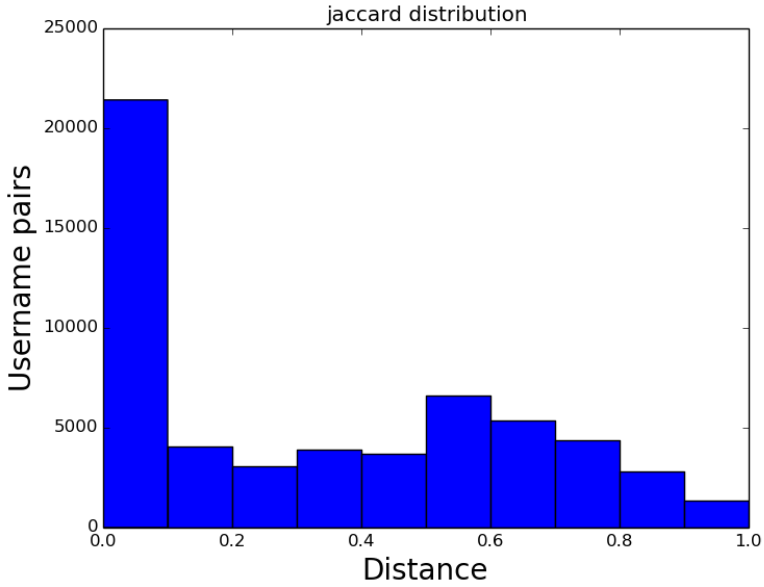
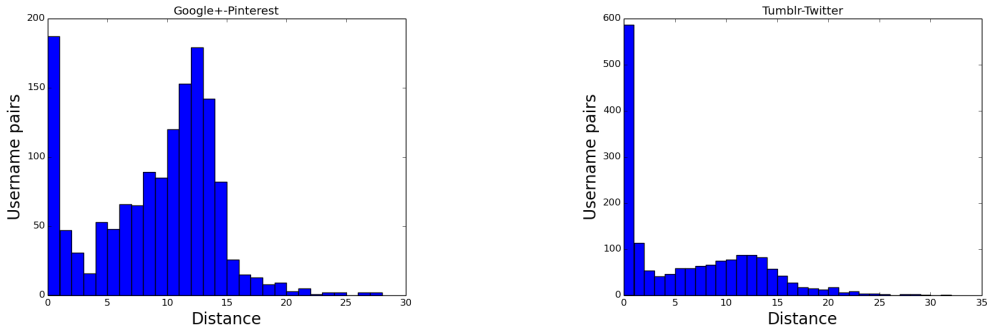


Figura 2: Distribuzione indice di Jaccard tra coppie di usernames



(a) Google+ Pinterest (b) Tumblr Twitter

Figura 3: Distribuzione bimodale, con pattern a doppio picco, anche su classe singola

In Figura 3 mostriamo come i dati riportino un pattern a *doppio picco* sia su dati globali che su classi prese singolarmente.

In Figura 4 e 5 mostriamo la distribuzione delle distanze tra coppie di username casuali, non appartenenti alla stessa persona. Il mescolamento delle coppie di username é ottenuto tramite una tecnica usualmente riferita come *convolution* o *zip function*, una funzione che mappa una tupla di sequenze in una sequenza di tuple. Se ad esempio si ha una lista di 3 coppie/tuple di username u per utenti U:

$$[(U_1u_1, U_1u_2), (U_2u_1, U_2u_2), (U_3u_1, U_3u_2)]$$

otteniamo, tramite l'inverso della funzione descritta prima (*unzip*), una lista con due tuple:

$$[(U_1u_1, U_2u_1, U_3u_1), (U_1u_2, U_2u_2, U_3u_2)]$$

A questo punto, mescoliamo in maniera casuale l'ordine degli elementi nelle due tuple, e attraverso la funzione zip ricombiniamo le due tuple, ottenendo cosi coppie di username non appartenenti allo stesso utente U. Ad esempio potremmo ritrovare:

$$[(U_1u_1, U_2u_2), (U_2u_1, U_3u_2), (U_3u_1, U_1u_2)]$$

Applichiamo questa tecnica alle coppie di username del nostro dataset, e calcoliamo la distribuzione delle distanze in maniera analoga a quella presentata precedentemente. Quello che riscontriamo é, una piú prevedibile, distribuzione normale (Gaussiana), come si puó notare in figure 4 e 5.

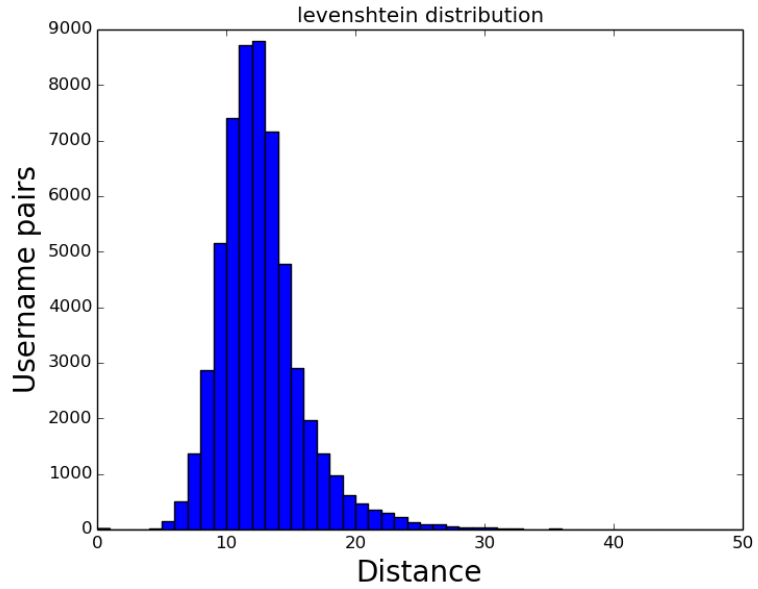


Figura 4: Distribuzione unimodale distanza di Levensthein tra coppie random

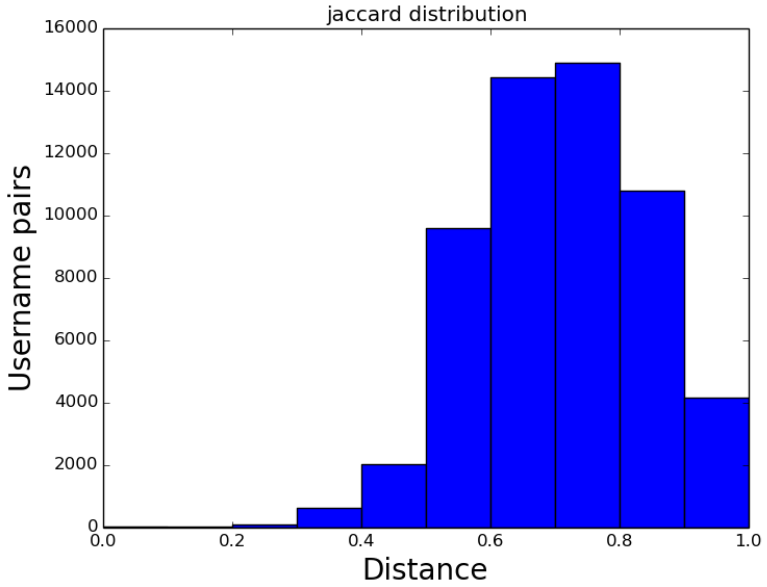


Figura 5: Distribuzione indice di Jaccard tra coppie random

Capitolo 4

Risultati

Performance classificatore.

Bibliografia

[1] M. Gotti, I linguaggi specialistici, Firenze, La Nuova Italia, 1991.

[2] R. Wellek, A. Warren, Theory of Literature , 3rd edition, New York, Harcourt, 1962.

[3] A. Canziani et al., Come comunica il teatro: dal testo alla scena. Milano, Il Formichiere, 1978.

[4] Ministry of Defence, Great Britain, Author and Subject Catalogues of the Naval Library, London, Ministry of Defence, HMSO, 1967.

[5] H. Heine, Pensieri e ghiribizzi. A cura di A. Meozzi. Lanciano, Carabba, 1923.

[6] L. Basso, "Capitalismo monopolistico e strategia operaia", Problemi del socialismo, vol. 8, n. 5, pp. 585-612, 1962.

[7] L. Avirovic, J. Dodds (a cura di), Atti del Convegno internazionale "Umberto Eco, Claudio Magris. Autori e traduttori a confronto" (Trieste, 27-28 novembre 1989), Udine, Campanotto, 1993.

[8] E.L. Gans, "The Discovery of Illusion: Flaubert's Early Works, 1835-1837", unpublished Ph.D. Dissertation, Johns Hopkins University, 1967.

[9] R. Harrison, Bibliography of planned languages (excluding Esperanto). `http://www.vor.nu/langlab/bibliog.html`, 1992, agg. 1997.