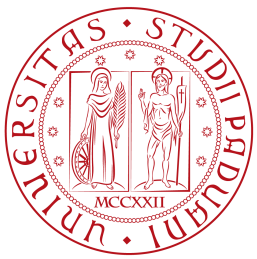
Università degli Studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea triennale in
Statistica per le Tecnologie e le Scienze



RELAZIONE FINALE

Confronto dell'impatto dei principali stemmer sull'efficacia di un sistema di Information Retrieval

Relatore: Prof. Massimo Melucci
Dipartimento di Ingegneria dell'Informazione

Laureando: Riccardo Baratto
Matricola: 1218761

Anno accademico: 2022/2023

Indice

Capitolo 1

Introduzione

1.1 Abstract tesi

Nel linguaggio naturale, le parole possono subire delle variazioni dovute a diversi fattori, queste variazioni in linguistica vengono identificate come flessioni. Lo stemming è il processo che riduce una forma flessa di una parola alla sua forma radice, il che può essere cruciale nell'ambito dell'Information Retrieval. Basandosi sull'assunzione che termini che condividono la radici hanno solitamente un significato simile, il processo di stemming viene ampiamente utilizzato nell'Information Retrieval[1] per migliorare le performance nel reperimento. L'obiettivo di questa tesi è analizzare e confrontare l'impatto di diversi algoritmi di stemming sull'efficacia di un sistema di Information Retrieval, basandosi su una serie di metriche mirate a valutare gli effetti degli stemmer sul recupero di documenti. Utilizzando la collezione TREC 2004 Robust, in combinazione con Elasticsearch e Python, è stata condotta un'analisi per determinare se e quali stemmer influiscono positivamente sull'efficacia del reperimento.

[1] o IR

1.2 Argomenti trattati

Nel mondo di oggi, la ricerca testuale è diventata di massima importanza, e l'accesso rapido e accurato alle informazioni è cruciale. In questo contesto, gioca un ruolo fondamentale l'Information Retrieval (IR), che rappresenta l'insieme dei processi utilizzati per selezionare, all'interno di un dataset, le informazioni rilevanti o utili in relazione a un particolare bisogno informativo. Questa disciplina costituisce la base dei motori di ricerca, ma questi motori non possono interpretare il bisogno informativo dell'utente se non attraverso un'interrogazione[2] formulata in un linguaggio che il sistema può elaborare. Nel linguaggio naturale, le parole possono subire variazioni, dovute a diversi fattori, identificate in linguistica come flessioni [1]. Lo stemming è il processo che riduce una forma flessa di una parola alla sua forma radicale [2], il che può risultare cruciale nell'ambito dell'Information Retrieval. Basandosi sull'assunzione che termini che condividono la stessa radice abbiano solitamente un significato simile, il processo di stemming è ampiamente utilizzato nell'Information Retrieval per migliorare le performance nel recupero delle informazioni [3]. Gli algoritmi di stemming possono risultare utili in diversi compiti di elaborazione del linguaggio naturale, come la classificazione dei testi e il recupero delle informazioni [4]. Tuttavia, lo stemming può anche avere effetti negativi, come la produzione di radici non reali. Esistono vari algoritmi di stemming, e l'obiettivo principale di questa tesi sarà il confronto tra questi algoritmi per valutare il loro impatto sull'efficacia di un sistema di Information Retrieval e le eventuali limitazioni.

[2]successivamente verrà utilizzata la parola query

1.3 Obiettivo della tesi

L'obiettivo principale di questa tesi è condurre un'analisi dettagliata e comparativa di diversi algoritmi di stemming per valutare il loro impatto sull'efficacia di un sistema di Information Retrieval. Saranno presi in considerazione vari aspetti degli algoritmi, nonché il loro influsso sulla capacità del sistema di recuperare documenti pertinenti. L'analisi mira a determinare se l'uso di un algoritmo di stemming migliora l'efficacia del recupero. Durante questa tesi, saranno utilizzate diverse metriche per valutare l'efficacia dei diversi stemmer nell'ambito del recupero dei documenti. Queste metriche includeranno, ma non saranno limitate a, MAP (Mean Average Precision), R-precision, e altre metriche mirate a valutare la rilevanza dei risultati di ricerca. L'obiettivo è comprendere come ciascun algoritmo di stemming influisca su queste metriche e come ciò si rifletta sulla qualità del recupero dei documenti. Per ottenere una comprensione completa dell'impatto degli algoritmi di stemming, verrà condotta una valutazione sia intrinseca che estrinseca. La valutazione intrinseca si concentrerà sulla misurazione diretta delle prestazioni degli stemmer su un vocabolario di parole generato dai documenti che compongono la collezione di test. Allo stesso tempo, la valutazione estrinseca esaminerà come questi stemmer influenzano il funzionamento del sistema di Information Retrieval in un contesto di utilizzo del mondo reale. È importante valutare entrambi questi aspetti degli algoritmi poiché sono strettamente correlati. Come collezione di test, è stata scelta la collezione sperimentale TREC Robust 2004, per la quale sono disponibili documenti, query e giudizi di rilevanza.

1.4 Struttura della tesi

Questo documento è sviluppato nei seguenti capitoli, di cui si riporta una breve presentazione:

- Fondamenti teorici
  - Nel corso di questo capitolo, esploreremo i concetti fondamentali relativi all'indicizzazione e alla ricerca testuale. Approfondiremo il concetto di stemming e la sua importanza nell'Information Retrieval, spiegando come gli algoritmi di stemming vengano utilizzati per ridurre le parole alla loro forma radicale. Questo permette al sistema di ricerca di considerare le variazioni linguistiche e di recuperare documenti pertinenti anche quando le forme delle parole differiscono. Alla fine del capitolo, esamineremo i limiti e le criticità degli algoritmi di stemming. Questo capitolo costituirà una base teorica solida per il resto della tesi, fornendo una comprensione chiara dei concetti chiave ad essa correlati.
- Valutazione della forza degli algoritmi di stemming
  - Durante questo capitolo, definiremo il corpus di test e illustreremo i metodi utilizzati per valutare l'efficacia degli algoritmi di stemming selezionati. Successivamente, calcoleremo diverse metriche che consentiranno di analizzare le capacità di ciascuno stemmer. Nell'ultima parte del capitolo, presenteremo i risultati ottenuti e li discuteremo in dettaglio.
- Confronto degli algoritmi di stemming nella ricerca dell'informazione
  - Lo scopo di questo capitolo è descrivere in dettaglio la procedura utilizzata per condurre questa comparazione. Inizieremo confrontando le query della stessa lunghezza ma relative a topic diversi, per poi affrontare un caso più realistico in cui verranno utilizzati tutti i topic. Nell'ultima parte del capitolo, presenteremo i risultati e condurremo un test di significatività, in particolare il test di Wilcoxon, al fine di valutarne la rilevanza statistica.

- Tabelle
  - In questo capitolo verranno inserite le tabelle contenenti le regole, le condizioni e i suffissi dei vari algoritmi di stemming selezionati.

Capitolo 2

Fondamenti teorici

2.1 Concetti di indicizzazione

L'indicizzazione è strutturata in step, alcuni obbligatori, altri facoltativi eseguiti a seconda degli obiettivi di progettazione del sistema; gli step sono i seguenti:

- Alimentazione della collezione
  - L'alimentazione della collezione può avvenire in modo manuale, semi-automatico o automatico, in base al controllo effettuato da personale specializzato.
- Analisi lessicale
  - É il procedimento con cui il testo viene suddiviso per rilevare ed estrarre le stringhe che sono potenzialmente termini o parole chiave. Le parole vengono rilevate a ogni carattere di separazione come simboli di punteggiatura e spazi.

  a sua volta articolata in:
  - Rimozione delle stop word
  - Stemming
  - Costruzione dei termini
- Implementazione dell'indice

2.2 Concetto di stemming e la sua importanza

Nel linguaggio naturale, esistono molte parole che esprimono lo stesso significato, il che può rappresentare una sfida nel reperimento delle informazioni poiché il reperimento di documenti pertinenti si basa sulla corrispondenza delle parole. Per affrontare questo problema, invece di limitare le corrispondenze alle parole identiche, esistono numerose tecniche che consentono di associare parole semanticamente correlate. Come anticipato, lo stemming è un processo che riduce una parola alla sua radice o tema. In modo più preciso, lo stemming riduce le diverse forme di una parola derivanti dall'inflessione (come i plurali o i tempi verbali) o dalla derivazione (come la trasformazione di un verbo in un sostantivo) [1] a una radice comune. Questa procedura è spesso utilizzata durante la fase di preparazione del testo per analisi successive. Nell'Information Retrieval (IR), lo stemming garantisce la corrispondenza tra le varianti di una parola durante la ricerca [2]. Ad esempio, le parole "walking" e "walked" possono essere ridotte alla stessa radice "walk". Una volta che queste parole sono state ridotte alla radice, l'occorrenza di una delle due parole corrisponderà all'altra nella ricerca, migliorando così l'efficacia del reperimento delle informazioni.

2.3 Limiti degli algoritmi di stemming

L'utilizzo dello stemming presenta diverse problematiche. I principali errori che si possono verificare sono i seguenti:
Under stemming errors (USE): L'under stemming si verifica quando due o più parole vengono ridotte a più di una radice, quando in realtà dovrebbero essere ridotte alla stessa radice. Ad esempio le parole "data" e "datum" con alcuni algoritmi vengono ridotti rispettivamente a "dat" e "datu", quando entrambe dovrebbero essere riportate alla radice "dat"

Over stemming errors (OSE): L'over stemming si verifica quando due o più parole vengono ridotte alla stessa radice quando, invece, dovrebbero essere portate a due radici differenti. Un esempio sono le parole "university" e "universe" che vengono portate alla radice "univers".

Inoltre, le regole su cui si basano gli algoritmi di stemming "Rule Based"[1] cambiano in base alla lingua. Questo può essere un problema in quanto trattando dei documenti con più di una lingua al suo interno (per esempio un documento in lingua italiana, ma con dei termini in inglese) l'efficacia dello stemming può venir meno.

[1] La categoria degli stemmer che verranno confrontati in questa tesi

2.4 Algoritmi di stemming

2.4.1 Algoritmo di Porter

L'algoritmo di Porter [5] è stato presentato la prima volta nel 1979 presso il Computer Laboratory, Cambridge(Regno Unito), come parte di un progretto di IR più grande. Prima di esporre l'algoritmo bisogna introdurre la definizione di consonante secondo Porter. Una consonante è una lettera diversa da A, E, I, O e U. Viene considerata consonante anche la lettera Y, se preceduta da una vocale. Facendo un esempio, in "TOY" vengono considerate consonanti T e Y, mentre considerando la parola "SYZYGY" vengono ritenute consonanti le lettere "S", "Z" e "G". D'ora in poi le consonanti verranno indicate con la lettera "c", mentre le vocali con "v". Una lista di consonanti consecutive di lunghezza maggiore di 0 verrà indicata invece con "C", mentre una lista di vocali con le stesse caratteristiche verrà segnalata con "V". Tramite questa codifica ogni parola(o parte di essa) può essere codificata nella seguente forma: [C]VCVC...[V], dove la parte tra parentesi quadre indica la presenza arbitraria del loro contenuto. La forma [C](VC)m[V] invece indica la ripetizione di (VC) m volte, grazie a m = 0 si può includere la casistica della parola nulla. Le regole introdotte da questo algoritmo di rimozione dei suffissi vengono riportate nella seguente formula: (condizione) S1 -> S2, questo significa che se il suffisso S1 rispetta la condizione indicata verrà sostituita con S2. [5] La condizioni possono essere una o più delle seguenti:

- *S - la radice risultante termina con "S" (variando la lettera maiuscola varia anche la lettere con cui terminerà la parola)
- *v* - la radice risultante contiene con una vocale
- *d - la radice risultante termina con una doppia consonante
- *o - la radice risultante termina con la seguenza cvc, dove la seconda consonante non è "W", "X" o "Y"

Questo algoritmo si basa su 5 diversi step, nei quali vengono poste diverse regole, che verranno applicate in ordine, così da rimuovere il suffisso di lunghezza maggiore. Nella sezione A.1 verrà riportata la struttura di questo algoritmo.

2.4.2 Algoritmo di Porter 2

Questo algoritmo è stato sviluppato sempre da M.F Porter nel 2001 [6], basandosi sul suo primo algoritmo cerca di correggere alcuni errori aggiungendo delle nuove regole o cambiando quelle implementate precedentemente. I cambiamenti principali effettuati sono i seguenti:

- Se "y" è la lettera finale di una parola, viene cambiata in "i" meno spesso.
- Il suffisso "us" non perde più la lettera "s"
- Sono stati aggiunti alcuni suffissi, come "ly"
- È stata inserita una lista di forme inusuali
- Gli step 5a e 5b dell'algoritmo di Porter originale sono stati unificati in un unico step, ciò comporta che il raddoppio della "ll" finale non avviene con la rimozione della lettera "e"
- Nello step 3 il suffisso "ative" viene rimosso solo se si colloca nella regione R2[2]
- É stato aggiunto uno step 0 per la gestione dell'apostrofo.

Effettuando un confronto tra il primo algortimo di Porter e questo su un set di parola di esempio, si è visto che vi è una differenza del poco meno del 5% a livello di parole stemmate [6] Nella sezione A.2 verrà riportata la struttura dell'algoritmo di Porter 2.

[2]R2 è la regione dopo la prima consonante che segue una vocale, o è la regione nulla alla fine della parola se non c'è una tale consonante.

2.4.3 Algoritmo di Lovins

É il primo algoritmo di stemming mai pubblicato [7], è stato sviluppato da Lovins J.B. nel 1968. La progettazione di questo algoritmo è stata influenzata dal vocabolario tecnico con cui Lovins si trovato a lavorare (vi è infatti una grande presenza di termini correlati a documenti nel campo dell'ingegneria dei materiali). Questa influenza ha avuto il risultato di limitare la presenza suffissi (ad esempio alcune delle terminazione più comuni non sono state inserite). Questo algoritmo è composto da 29 condizioni, 35 regole di trasformazione 294 diverse terminazioni. L'algoritmo di Lovins risulta essere più grande dell'algoritmo di Porter, in quanto ha una lista di suffissi più estesa, nonostante ciò risulta essere più veloce, questo anche poichè richiede solamente due fasi per la rimozione dei suffissi.

- Step 1:
  - Nel primo step, viene trovato la terminazione più lunga che soddisfa la condizione ad esso associata, dopodichè viene rimossa.
- Step 2:
  - Nel secondo step si applicano le regole per trasformare le terminazioni, questo passo viene eseguito indipendentemente dal fatto che la terminazione venga rimossa o meno nel primo step.

Nella sezione A.3 verrà riportata la struttura dell'algoritmo di Lovins.

2.4.4 Algoritmo di Paice/Husk

L'algoritmo di stemming di Paice/Husk è stato pubblicato nel 1990 da Chris D. Paice in collaborazione con Husk [8]. É un algoritmo iterativo basato su una tabella di regole. Le regole sono raggruppate in base alla lettera finale del suffisso, questo permette un rapido accesso alle regole nella tabella. Ogni regola rimuove o sostituisce un suffisso da una parola, ed è composta da 5 step, dei quali due sono opzionali. Le regole sono indicizzate secondo l'ultima lettera della terminazione per consentire una ricerca efficiente e sono descritte con le seguenti forme:

- Le terminazioni sono riportate in ordine inverso
- L'asterisco indica che la trasformazione avviene solo se la parola non è già stata trasformata
- Un numero indica il numero totale di caratteri che vengono rimossi dalla parola originale
- Può essere presente una stringa che sostituisce la terminazione
- Il simbolo ">" indica che l'iterazione continua, mentre "." la ferma

Nella sezione A.4 verranno riportate le regole dell'algoritmo di Paice/Husk.

2.4.5 Algoritmo di Harmann

L'algoritmo di Harmann, anche chiamato "S", è stato presentato per la prima volta nel 1991 [9]. Questo algoritmo è comunemente utilizzato per effettuare uno stemming minimo. Le regole per questo algoritmo si applicano solamente se la parola originale è composta da almeno tre caratteri e vengono applicate in modo dipendente dall'ordine, cioè la prima regola incontrata, che risulta applicabile è l'unica che verrà utilizzata.

- Se la parola termina con "ies", ma non con "eies" o "aies" allora "ies" ->"y"
- Se la parola termina con "es", ma non con "aes", "ees" o "oes" allora "es" -> "e"
- Se la parola termina con "s", ma non con "us" o "ss" allora "s" viene rimossa

2.4.6 Algoritmo per la rimozione dei prefissi e rimozione affissi con Porter

A differenza degli altri algoritmi presentati finora questo algoritmo[3] effettuerà rimozione sia degli suffissi che dei prefissi. La prima fase dell'algoritmo coinvolge la rimozione dei prefissi. Inizia verificando se la parola inizia con uno dei prefissi presenti in lista:

Tabella 2.1: Elenco dei prefissi

sym	an	end	am	self	infra
de	per	oc	ana	ecto	multi
intra	ob	dis	ag	counter	endo
retro	apo	macro	mis	neo	com
hetero	anti	omni	up	micro	un
supra	maxi	circum	ac	be	homo
ultra	co	ir	uni	over	a
auto	im	trans	bio	meta	tri
post	hemi	in	mid	ante	down
abs	mega	eco	iso	pre	il
di	syn	cata	mini	af	extra
epi	contra	con	fore	ad	super
dia	sub	semi	dys	al	em
re	ex	arch	mono	of	ab
tele	under	hyper	para	non	ec
bi	op	mal	pro	en	inter
paleo	out	peri			

se presente questo viene rimosso solamente se la lunghezza della parola dopo aver rimosso il prefisso è maggiore di 0, in caso questa condizione non venisse soddisfatta non verrà effettuata la rimozione del prefisso. Successivamente, sia se la rimozione del prefisso è avvenuta o meno, verrà applicato l'algoritmo di Porter per effettuare la rimozione di un possibile suffisso.

[3]Successivamente verrà chiamato Affix-Porter

2.4.7 Algoritmo di rimozione prefissi e suffissi

Algoritmo simile a quello visto in precedenza, la rimozione dei prefissi avviene allo stesso modo e con la stessa lista di prefissi. Per la rimozione dei suffissi invece non si utilizzerà più l'algoritmo di Porter, ma bensì verrà utilizzata una procedura simile a quella usata per la rimozione dei prefissi, impiegando però una lista di suffissi.[4] I suffissi utilizzati sono i seguenti: Le

Tabella 2.2: Elenco dei Suffissi Inglesi

able	al	ance	ancy	ant	ary	ate
en	er	ful	ic	ify	ing	ion
ism	ist	ity	ive	ize	less	ly
ment	ness	ous	ship	sion	tion	ty
ward	wards	wise	able	ible	al	ial
an	ian	ance	ence	ant	ent	ar
ary	ate	dom	ee	eer	er	or
ese	ess	ful	hood	ic	ical	ion
ish	ism	ist	ite	ity	ive	less
like	ly	ment	ness	ous	ship	some
sion	tian	tion	ty	ward	wise	y
ed						

differenze sostanziali con l'algoritmo precedente sono il numero di suffissi e le regole per cui vengono rimossi.

[4] Successivamente verrà indicato come Pre-Suf

2.4.8 Algoritmo misto

Questo algoritmo è una combinazione degli stemmer più famosi:

- Porter
- Porter 2
- Paice/Husk
- Lovins

Consiste nel ridurre una parola alla radice più corta tra quelle prodotte da questi algoritmi. In particolare il 2.3% delle parole sono state stemmate con l'algoritmo di Porter, il 2.4% con l'algoritmo di Porter 2, il 44.5% con l'algoritmo di Lovins mentre il restante 50.8% è stato stemmato con l'algoritmo di Paice/Husk . Lo scopo di questo algoritmo in questa tesi è quello di vedere se enfatizzando lo stemming vi è un impatto nell'efficacia nel reperimento.

2.5 Definizione delle metriche di valutazione

La qualità di uno stemmer può essere valutata in due modi:

- Quanto correttamente lo stemmer riporta parole semanticamente e morfologicamente correlate alla stessa radice
- Quanti miglioramenti porta lo stemmer all'Information Retrieval

In accordo con Jones and Galliers [10] e con Mollà and Hutchinson [11] il primo modo è una valutazione intrenseca in quanto valuta l'accuratezza dello stemmer come sistema autonomo. Il secondo modo è una valutazione estrinseca in quanto analizza il suo impatto in una delle sue applicazioni, come il reperimento.

Capitolo 3

Valutazione degli algoritmi di stemming

3.1 Corpus di test

La collezione utilizzata per effettuare questo confronto è TREC[1] Robust 2004. Questa collezione è composta da circa mezzo milione di documenti tratti dal Finalcial Times, il Federal Register, il LA Times e il FBIS. Per l'esattezza

Sorgente	Numero documenti	Peso (MB)
Financial Times	210,158	564
Federal Register 94	55,630	395
FBIS, disk 5	130,471	470
LA Times	131,896	475

Questa collezione è conosciuta per contenere molte interrogazioni difficili, che costituiscono una sfida anche per i motori di ricerca odierni. Per questo viene spesso usata per le ricerche nel settore. I topic considerati difficili sono i seguenti:[12]

303	322	344	353	363	378	394	408	426	439
307	325	345	354	367	379	397	409	427	442
310	330	346	355	372	383	399	414	433	443
314	347	356	374	389	401	416	435	445	442
320	341	350	362	375	393	404	419	436	448

[1] Text REtrieval Conference

3.2 Procedura di valutazione comparativa

Per effettuare una valutazione intrinseca si è pensato di utilizzare le seguenti metriche:

- Il numero medio di parole per conflation class(MWCC)
  - Questa metrica misura il numero di parole differenti che vengono stemmizzate nella stessa radice. Per esempio se le parole "runner", "running" e "run" vengono tutte stemmate a "run", allora la conflation class per la radice "run" è 3. Stemmer considerati più forti tendono ad avere una conflation class media più alta
- Index compression factor(ICF)
  - Questa metrica misura di quanto viene ridotto l'indice attraverso lo stemming. Stemmer più forti tendono ad avere un index compression factor maggiore.
- Numero medio di caratteri rimossi in ogni parola(MCR)
  - Stemmer più forti tendono a rimuovere più caratteri.
- La media della distanza di Hamming modificata tra le parole e le loro radici(MDHM)
  - La distanza di Hamming solitamente misura la distanza tra due stringhe di uguale lunghezza, per parole di lunghezza diversa è stata aggiunta la differenza in lunghezza tra le due parole alla distanza di Hamming originale. Così facendo consente di tenere conto delle trasformazioni effettuate dallo stemming. [13]
- Numero parole uniche post stemming (NPUPost)
- Numero medio caratteri post stemming (NMCPost)

Queste misure solitamente vengono effettuate su un vocabolario di paorle uniche. Nel caso di Frakes [13] è stato utilizzato un vocabolario contente 49656 parole provenienti dall'UNIX spelling dictionary e dal corpus Moby. Per ottenere un'analisi coerente sono state prese tutte le parole uniche presenti nei documenti della collezione TREC Robust 2004.

3.3 Valutazione intrinseca

La forza di uno stemmer può risultare predittiva riguardo al richiamo e alla precisione nel reperimento dei documenti. Uno stemmer più forte, in media, aumenterà il richiamo, diminuirà la precisione e avrà un Index compression factor maggiore.

Indice	**ICF**	**MWCC**	**MCR**	**MDHM**	**NPUPost**	**NMCPost**
Lovins	30.30%	1.43	1.28	1.32	285310	6.60
Affix-Porter	26.60%	1.36	1.22	2.47	300446	6.65
Pre-Suf	16.05%	1.19	0.97	2.24	343633	6.90
Porter	20.00%	1.25	0.79	0.85	327456	7.08
S	8.92%	1.10	0.14	0.15	372783	7.73
Porter 2	20.04%	1.25	0.80	0.83	327265	7.08
Paice/Husk	32.15%	1.47	1.50	1.53	277721	6.37
Misto	36.47%	1.57	1.71	1.75	260020	6.17

Tabella 3.1: Metriche valutazione forza degli stemmer

3.4 Risultati

Basandosi su queste misure si può identificare una classifica di questi algoritmi dal più forte al meno. La classifica è la seguente:

- Misto
- Paice/Husk
- Lovins
- Rimozione affissi e dopo Porter
- Porter
- Porter 2
- Rimozione affissi e suffissi
- S

La forza di uno stemmer è determinata dalla sua aggressività nel troncare le parole. Gli stemmer più forti tendono a produrre radici più corte e hanno una maggiore tendenza a eliminare affissi e desinenze. Al contrario, gli stemmer meno forti mantengono più informazioni delle parole originali. [13] La forza degli stemmer ha un impatto diretto sulla precisione e sul richiamo di un sistema di reperimento. Gli stemmer meno aggressivi conservano più informazioni delle parole originali, poichè effettuano uno stemming più leggero rispetto agli stemmer più aggressivi. Questo implica una minor perdita di precisione rispetto che agli algoritmi più aggressivi, ma anche un minor guadagno in richiamo.[13] Al contrario, gli stemmer più aggressivi, aumentano in media maggiormente il richiamo, comportando però un decremento maggiore della precisione. Come ci si poteva aspettare l'algoritmo misto risulta essere quello più aggressivo, seguito poi Paice/Husk e Lovins. Quest'ultimi due algoritmi ottengono risultati simili, stessa cosa per l'algoritmo di Porter e di Porter 2.

Capitolo 4

Impatto degli algoritmi di stemming sull'efficacia di reperimento dell'informazione

4.1 Descrizione dell'esperimento

La collezione di documenti utilizzata[12] è accompagnata con un set di query che vengono utilizzate per la valutazione del reperimento. Ogni query è una descrizione di un bisogno informativo, costruita in linguaggio naturale. Queste query hanno una struttura come la seguente:

Number 301
Title International Organized Crime
Description Identify organizations that participate in international criminal activity, the activity, and, if possible, collaborating organizations and the countries involved.
Narrative A relevant document must as a minimum identify the organization and the type of illegal activity (e.g., Columbian cartel exporting cocaine). Vague references to international drug trade without identification of the organization(s) involved would not be relevant.

Per avere una visione completa nel reperimento dell'informazione si è deciso di utilizzare nove combinazione di query, in particolare abbiamo:

- Query effettuate solo con "title" solamente su topic considerati "non difficili" - 200 query

- Query effettuate solo con ”title” solamente su topic considerati ”difficili” - 50 query
- Query effettuate solo con ”desc” solamente su topic considerati ”non difficili” - 200 query
- Query effettuate solo con ”desc” solamente su topic considerati ”difficili” - 50 query
- Query effettuate con una combinazione di ”title” e ”desc” solamente su topic considerati ”non difficili” - 200 query
- Query effettuate con una combinazione di ”title” e ”desc” solamente su topic considerati ”difficili” - 50 query
- Query effettuate solo con ”title” solamente su tutti i topic(”non difficili” e ”difficili”) - 250 query
- Query effettuate solo con ”desc” solamente su tutti i topic(”non difficili” e ”difficili”) - 250 query
- Query effettuate con una combinazione di ”title” e ”desc” su tutti i topic(”non difficili” e ”difficili”) - 250 query

Oltre al set di query, vengono forniti anche i giudizi di rilevanza, che consento di discernere se un dato documento risulta essere rilevante o meno per una determinata query. Durante la fase di reperimento le parole chiave delle query vengono abbinate alle parole chiave dei documenti, e i documenti rilevanti vengono recuperati in ordine decrescente di rango. Pertanto, in un sistema IR, le prestazioni di uno stemmer, relative all'effiacia nel reperimento, possono essere misurate attraverso le seguenti metriche:

- MAP: E' definita come la media delle precisioni medie non interpolate su un insieme di interrogazioni. La precisione media non interpolata viene determinata come AP, ed è la media delle precisioni calcolate ad ogni documenti rilevante trovato nel ranking. [14]
- R-Precision: L'R-Precision è definita come

$$\frac{r}{R}$$

ovvero il rapporto tra tutti i documenti rilevanti recuperati(r) e il numero di documenti rilevanti presenti(R).[1]

- F1-score: Questa metrica considera sia la precisione che il richiamo per verificare l'accuratezza del recupero. E' la media ponderata di precisione e richiamo.

$$F = \frac{2RP}{R+P}$$

Oltre a queste misure verrà riportato il numero totale di documenti rilevanti reperiti[2] Per poter valutare l'impatto di un algoritmo di stemming sull'efficacia in un sistema di reperimento occurre confrontare una baseline in cui non viene applicato lo stemming con un'esecuzione in cui viene applicato. Per poter calcolare le misure di efficacia del reperimento, è necessario utilizzare una raccolta di test di prova composta da documenti, query e i giudizi di rilevanza per ogni query. La raccolta test utilizzata, TREC Robust 2004, non include giudizi di rilevanza esaustivi su tutte le coppie query-documento. Nonostante ciò, Voorhess[15] ha concluso che vi è una correlazione molto alta tra le classifiche prodotte utizzando diversi set di giudizi di rilevanza. Ciò indica che la valutazione delle prestazione di recupero rimane stabile nonostante differenze sostanziali nei giudizi di rilevanza. In questi esperimenti di recupero, è stata valutata la significatività utilizzando il test di Wilcoxon imponendo un alpha = 0.05. Il test è stato effettuato in modo automatico utilizzando la funzione "wilcoxon" del modulo "stats"[3] della libreria "scipy". Per poter rappresentare la significatività e contemporaneamente dare una visione immediata dell'eventuale incremento o peggiormento rispetto alle baseline, successivamente alla tabella dei risultati numerici, verrà fornita una tabella con la seguente simbologia:

- =
  - Vi è questo simbolo quando la differenza nella metrica considerata non risulta significativa.

[1] Questa misura è stata introdotta per il TREC2 da Chris Buckley(Cornell University

[2] Per valutare la significatività di questa misura verrà utilizzato il numero di documenti rilevanti recuparati da ogni query

[3] https://docs.scipy.org/doc/scipy/reference/stats.html

- +
  - Vi è questo simbolo quando vi è una differenza significativa e la media della metrica presa in considerazione risulta essere maggiore nell'algoritmo di stemming considerato rispetto alla baseline.
- -
  - Vi è questo simbolo quando vi è una differenza significativa e la media della metrica presa in considerazione risulta essere minore nell'algoritmo di stemming considerato rispetto alla baseline.

4.2 Risultati e analisi dei dati

4.2.1 Risultati query Title su topic "Non difficili" e "Difficili"

In questa sezione andremo ad analizzare i risultati in cui si utilizzano query corte, cioè quelle in cui si utilizza solo il "Title". Verranno effettuate sui topic "Non difficili" e su quelli considerati "Difficili"

Nome indice	MAP	R-Precision	F1-Score	Numero documenti rilevanti recuperati
S	0.2306	0.4370	0.2261	3706.0
Affix-Porter	0.2054	0.4030	0.2093	3438.0
Pre-Suf	0.1789	0.3667	0.1933	3184.0
Porter 2	0.2348	0.4435	0.2292	3745.0
Paice/Husk	0.2195	0.4255	0.2183	3567.0
No stemming	0.2085	0.4066	0.2104	3447.0
Porter	0.2329	0.4403	0.2280	3725.0
Lovins	0.2077	0.4052	0.2098	3473.0
Misto	0.2049	0.4042	0.2080	3433.0

Tabella 4.1: Title - non difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	+	+	+	+
Affix-Porter	=	=	=	=
Pre-Suf	-	-	-	-
Porter 2	+	+	+	+
Paice/Husk	=	=	=	=
Porter	+	+	+	+
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.2: Title - non difficili Significarività

Da questi risultati possono esseere tratte diverse conclusioni in base all'algoritmo di stemming utilizzato:

- S:

  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 10.59% per il MAP, del 7.48% per l'R-precision, del 7.44% nell'F-score e vengono reperiti 259 documenti rilevanti in più rispetto alla baseline

- Affix-Porter
  - Con questo algoritmo non si osservano miglioramenti o peggioramenti significativi rispetto alla baseline.

- Pre-Suf
  - Con questo algoritmo si osservano peggioramenti significativi in ogni misura. Il MAP ha un peggioramento di 14.17%, l'R-Precision di 9.82% e di 8.11% nell'F1-Score. Inoltre vengono reperiti 263 documenti rilevanti in meno.

- Porter 2
  - Con l'algoritmo "Porter 2", si osserva un miglioramento significativo in ogni misura, simile a quanto visto con "S". Il MAP è aumentato significativamente di 12.58%, l'R-precision di 9.08%, l'F-score aumenta di 8.94% e vengono reperiti 298 documenti rilevanti in più rispetto alla baseline.

- Paice/Husk
  - L'algoritmo di Paice/Husk non ha portato dei miglioramenti o peggioramenti significativi in nessuna misura.

- Porter
  - Con l'algoritmo "Porter", si osserva un miglioramento significativo in ogni misura, simile a quanto visto con "S". Il MAP è aumentato significativamente di 11.71%, l'R-precision di 8.28%, l'F-score aumenta di 8.38% e vengono reperiti 278 documenti rilevanti in più rispetto alla baseline.

- Lovins

- L'algoritmo "Lovins" non ha mostrato miglioramenti significativi rispetto alla baseline "No Stemming" in nessuna delle misure.

- Misto
  - L'algoritmo "misto" non ha mostrato miglioramenti significativi rispetto alla baseline "No Stemming" in nessuna delle misure.

In generale, per le query corte effettuate su topic considerati "non difficili", i risultati indicano che l'S, Porter 2 e Porter sono gli unici algoritmi che portano un miglioramento significativo all'efficacia del reperimento. Al contrario osserviamo come l'algoritmo Pre-Suf porta dei peggioramenti nell'efficacia.

Nome indice	MAP	R-Precision	F1-Score	Numero documenti rilevanti recuperati
S	0.0726	0.2244	0.1359	605.0
Affix-Porter	0.0534	0.1890	0.1138	526.0
Pre-Suf	0.0557	0.1913	0.1158	546.0
Porter 2	0.0623	0.2107	0.1287	615.0
Paice/Husk	0.0649	0.2088	0.1285	613.0
No stemming	0.0720	0.2221	0.1346	612.0
Porter	0.0619	0.2105	0.1287	615.0
Lovins	0.0646	0.2133	0.1246	577.0
Misto	0.0627	0.2037	0.1216	573.0

Tabella 4.3: Title - Difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	=	=	=	=
Affix-Porter	=	-	=	=
Pre-Suf	-	-	=	-
Porter 2	=	=	=	=
Paice/Husk	=	=	=	=
Porter	=	=	=	=
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.4: Title - Difficili Significatività

Effettuando le query solo su topic considerati 'difficili' i risultati risultano differenti:

- S:
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.
- Affix-Porter
  - Questo algoritmo non ha portato dei miglioramenti o peggioramenti significativi in nessuna misura ad eccezione che per l'R-Precision, in cui si nota un peggioramento significativo del 14.91%

- Pre-Suf
  - Questo algoritmo non ha portato dei miglioramenti o peggioramenti significativi nell'F1-Score. Nelle altre misure, invece, si può notare dei peggioramenti significativi. Nel MAP vi è un peggioramento del 22.71%, del 13.82% nel R-Precision e vengono reperiti 66 documenti rilevanti in meno.
- Porter 2
  - Con l'algoritmo Porter 2 non si osserva alcuna differenza significativa rispetto alla baseline.
- Paice/Husk
  - Con questo algoritmo non si osserva alcuna differenza significativa rispetto alla baseline.
- Porter
  - Con l'algoritmo di Porter non si osserva alcuna differenza significativa rispetto alla baseline.
- Lovins
  - Con l'algoritmo di Lovins non si osserva alcuna differenza significativa rispetto alla baseline.
- Misto
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

I risultati suggeriscono che gli algoritmi di stemming selezionati possono influenzare in modo diverso le prestazioni nel recupero delle informazioni in base alla complessità del topic. S, Porter e Porter 2 sembrano avere un impatto significativo per i topic "non difficili", mentre nel caso dei topic considerati "difficili" nessun algoritmo si è dimostrato significativamente impattante in maniera positiva sull'efficacia nel reperimento, altrosì vi è un peggioramento significativo utilizzando i seguenti algoritmi: Affix-Porter e Pre-Suf.

4.2.2 Risultati query Desc su topic "Non difficili" e "Difficili"

In questa sezione andremo ad analizzare i risultati delle query di lunghezza media, cioè quelle in cui si utilizza solo il "Desc". Verranno effettuate sui topic "Non difficili" e su quelli considerati "Difficili" Da questi risul-

Nome indice	MAP	R-Precision	F1-Score	Numero documenti rilevanti recuperati
S	0.2224	0.4155	0.1968	3290.0
Affix-Porter	0.2060	0.3906	0.1884	3162.0
Pre-Suf	0.1933	0.3683	0.1788	2987.0
Porter 2	0.2281	0.4194	0.2015	3383.0
Paice/Husk	0.2066	0.3927	0.1912	3191.0
No stemming	0.2045	0.3851	0.1833	3041.0
Porter	0.2259	0.4204	0.2016	3383.0
Lovins	0.2094	0.3901	0.1894	3176.0
Misto	0.2024	0.3858	0.1864	3112.0

Tabella 4.5: Desc - non difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	+	+	+	+
Affix-Porter	=	=	=	=
Pre-Suf	-	-	-	-
Porter 2	+	+	+	+
Paice/Husk	=	=	=	=
Porter	+	+	+	+
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.6: Desc - non difficili Significatività

tati possiamo trarre diverse conclusioni, in base all'algoritmo di stemming utilizzato:

- S:
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare

vediamo un miglioramento del 8.77% per il MAP, del 7.89% per l'R-precision, del 7.36% nell'F-score e vengono reperiti 259 documenti rilevanti in più rispetto alla baseline.

- Affix-Porter
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.
- Pre-Suf
  - Con questo algoritmo di stemming possiamo notare un peggioramento significativo in ogni misura analizzata. In particolare vediamo un peggioramento del 5.48% per il MAP, del 4.37% per l'R-precision, del 2.45% nell'F-score e vengono reperiti 54 documenti rilevanti in meno rispetto alla baseline.
- Porter 2
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 11.59% per il MAP, del 8.92% per l'R-precision, del 9.91% nell'F-score e vengono reperiti 342 documenti rilevanti in più rispetto alla baseline.
- Paice/Husk
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.
- Porter
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 10.49% per il MAP, del 9.17% per l'R-precision, del 10.00% nell'F-score e vengono reperiti 342 documenti rilevanti in più rispetto alla baseline.
- Lovins
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

- Misto
    - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

Notiamo che gli algortmi di Porter, Porter 2 e S risultano avere un'impatto positivo nell'efficia di queste query. Il Pre-Suf comporta invece un impatto negativo. Gli altri stemmer invece non comportano miglioramenti o peggioramenti significativi.

Nome indice	MAP	R-Precision	F1-Score	Numero documenti rilevanti recuperati
S	0.0741	0.2268	0.1220	544.0
Affix-Porter	0.0627	0.2175	0.1126	496.0
Pre-Suf	0.0686	0.2115	0.1098	482.0
Porter 2	0.0695	0.2221	0.1205	544.0
Paice/Husk	0.0673	0.2178	0.1186	538.0
No stemming	0.0729	0.2138	0.1106	488.0
Porter	0.0702	0.2227	0.1204	544.0
Lovins	0.0679	0.2158	0.1118	494.0
Misto	0.0652	0.2150	0.1135	511.0

Tabella 4.7: Desc - Difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	=	=	=	=
Affix-Porter	=	=	=	=
Pre-Suf	=	=	=	=
Porter 2	=	=	=	=
Paice/Husk	=	=	=	=
Porter	=	=	=	=
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.8: Desc - Difficili Significatività

In questo caso nessun algoritmo di stemming ha un impatto significativo nell'efficacia del reperimento. Come in precedenza si può notare come la complessita della query abbia un'impatto sull'efficia degli algoritmi di stemming. Possiamo però notare come anche la lunghezza della query abbia il suo impatto, infatti si può notare come in base alla lunghezza della query vi è una differenza nella variazione percentuale rispetto alla baseline in ogni misura. Gli algoritmi "S", "Porter" e "Porter 2" sono stati gli unici ad avere un impatto significativo.

4.2.3 Risultati query Title + Desc su topic "Non difficili" e "Difficili"

Nome indice	MAP	R-Precision	F1-Score	Numero documenti rilevanti recuperati
S	0.2653	0.4760	0.2321	3935.0
Affix-Porter	0.2528	0.4517	0.2243	3826.0
Pre-Suf	0.2271	0.4231	0.2105	3576.0
Porter 2	0.2695	0.4824	0.2368	4031.0
Paice/Husk	0.2487	0.4586	0.2255	3801.0
No stemming	0.2436	0.4439	0.2157	3629.0
Porter	0.2678	0.4785	0.2352	3997.0
Lovins	0.2507	0.4509	0.2219	3769.0
Misto	0.2392	0.4416	0.2170	3676.0

Tabella 4.9: Title+Desc - non difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	+	+	+	+
Affix-Porter	+	+	+	+
Pre-Suf	-	=	=	=
Porter 2	+	+	+	+
Paice/Husk	=	+	+	+
Porter	+	+	+	+
Lovins	+	=	=	=
Misto	=	=	=	=

Tabella 4.10: Title+Desc - non difficili Significatività

Dai risultati delle query lunghe, cioè quelle in cui utilizziamo una combinazione di 'title' e 'desc', effettuate su topic considerati "non difficili" possiamo trarre diverse conclusioni, in base all'algoritmo di stemming utilizzato:

- S:
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 8.88% per il MAP, del 7.24% per l'R-precision, del 7.60% nell'F-score e vengono reperiti 306 documenti rilevanti in più rispetto alla baseline.

- Affix-Porter
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 3.77% per il MAP, del 1.76% per l'R-precision, del 3.97% nell'F-score e vengono reperiti 197.0 documenti rilevanti in più rispetto alla baseline.
- Pre-Suf
  - Con questo algoritmo osserviamo un decremento significativo MAP, mentre per le altre misure non vi è una differenza significativa. Vi è quindi, un decremento di 6.79% nel MAP
- Porter 2
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 10.66% per il MAP, del 8.68% per l'R-precision, del 9.78% nell'F-score e vengono reperiti 402 documenti rilevanti in più rispetto alla baseline.
- Paice/Husk
  - Con questo algoritmo non si notano differenze significative nel MAP, mentre per le altre misure si ottiene un guadagno significativo. Nell'R-Precision abbiamo un incremento del 3.31%, nell'F1-Score c'è un incremento del 4.53%, e vengono reperiti 172 documenti in più.
- Porter
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 9.93% per il MAP, del 7.77% per l'R-precision, del 9.03% nell'F-score e vengono reperiti 368 documenti rilevanti in più rispetto alla baseline.
- Lovins
  - Con questo algoritmo si ha una differenza significativa solo nel MAP, in particolare si ha un incremento del 2.90

- Misto
    - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

In questa casistica si può notare come alcuni algoritmi che fin'ora non aveva mostrato di essere impattanti nell'efficacia si sono rilevanti efficaci. Infatti, oltre agli algoritmi che portano incrementi significativi anche alle query corte e medie (quindi S, Porter e Porter 2), in questo caso si notano degli incrementi significativi anche in alcune misure per i seguenti algoritmi: Paice/Husk, Lovins e Affix-Porter.

Vediamo ora l'impatto dell'efficacia sui topic considerati difficili:

Nome indice	MAP	R-Precision	F1-Score	Numero documenti rilevanti recuperati
S	0.0895	0.2594	0.1462	674.0
Affix-Porter	0.0758	0.2504	0.1412	653.0
Pre-Suf	0.0840	0.2352	0.1366	633.0
Porter 2	0.0830	0.2594	0.1477	695.0
Paice/Husk	0.0830	0.2559	0.1480	693.0
No stemming	0.0918	0.2526	0.1390	643.0
Porter	0.0837	0.2627	0.1500	704.0
Lovins	0.0844	0.2519	0.1413	646.0
Misto	0.0834	0.2533	0.1438	666.0

Tabella 4.11: Title+Desc - Difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	=	=	=	=
Affix-Porter	=	=	=	=
Pre-Suf	=	=	=	=
Porter 2	=	=	=	=
Paice/Husk	=	=	=	=
Porter	=	=	=	=
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.12: Title+Desc - Difficili Significatività

In questo caso nessun algoritmo di stemming ha un impatto significativo nell'efficacia del reperimento. Come nei casi precedenti la difficoltà della query ha un impatto sull'efficacia degli stemmer, infatti nessun algoritmo ha un impatto significativo nel caso delle query fatte su topic difficili.

4.2.4 Risultati query su tutti i topic

Poichè nel mondo reale non vengono effettuate solamente query su topic non difficili o su topic difficili, si è deciso di effettuare tutte le query(quindi sia quelle difficili che non) per comprendere come lo stemming si comporta senza in una situazione più realistica.

Indice	MAP	R-Precision	F1-Score (beta=1)	Num. Doc. Rilevanti Recuperati
S	0.1994	0.3950	0.2083	4311.0
Affix-Porter	0.1754	0.3607	0.1904	3964.0
Pre-Suf	0.1546	0.3321	0.1780	3730.0
Porter 2	0.2007	0.3975	0.2093	4360.0
Paice/Husk	0.1889	0.3827	0.2005	4180.0
No stemming	0.1815	0.3702	0.1954	4059.0
Porter	0.1991	0.3949	0.2084	4340.0
Lovins	0.1794	0.3673	0.1930	4050.0
Misto	0.1768	0.3645	0.1909	4006.0

Tabella 4.13: Title - non difficili+Difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	+	+	+	+
Affix-Porter	=	=	=	=
Pre-Suf	-	-	-	-
Porter 2	+	+	+	+
Paice/Husk	=	=	=	=
Porter	+	+	+	+
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.14: Title - non difficili+Difficili Significatività

- S:
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 9.85% per il MAP, 6.69% per l'R-precision, del 6.60% nell'F-score e vengono reperiti 252.0 documenti rilevanti in più rispetto alla baseline.

- Affix-Porter
  - Con questo algoritmo non vi sono differenze significative.
- Pre-Suf
  - Con questo algoritmo di stemming possiamo notare un decremento significativo in ogni misura analizzata. In particolare vediamo un peggioramento del 14.80% per il MAP, 10.29% per l'R-precision, del 8.90% nell'F-score e vengono reperiti 329.0 documenti rilevanti in più rispetto alla baseline.
- Porter 2
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 10.57% per il MAP, 7.38% per l'R-precision, del 7.12% nell'F-score e vengono reperiti 301.0 documenti rilevanti in più rispetto alla baseline.
- Paice/Husk
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.
- Porter
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 9.68% per il MAP, 6.67% per l'R-precision, del 6.66% nell'F-score e vengono reperiti 281.0 documenti rilevanti in più rispetto alla baseline.
- Lovins
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.
- Misto
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

Per le query basate solo sul titolo, gli algoritmi S, Porter e Porter 2 sembrano trarre un notevole beneficio, mostrando miglioramenti significativi in tutte misure per entrambi i tipi di topic.

Indice	MAP	R-Precision	F1-Score (beta=1)	Num. Doc. Rilevanti Recuperati
S	0.1931	0.3782	0.1820	3834.0
Affix-Porter	0.1777	0.3564	0.1734	3658.0
Pre-Suf	0.1687	0.3373	0.1652	3469.0
Porter 2	0.1968	0.3804	0.1855	3927.0
Paice/Husk	0.1791	0.3581	0.1769	3729.0
No stemming	0.1785	0.3513	0.1689	3529.0
Porter	0.1951	0.3813	0.1856	3927.0
Lovins	0.1814	0.3557	0.1741	3670.0
Misto	0.1753	0.3520	0.1720	3623.0

Tabella 4.15: Desc - non difficili+Difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	+	+	+	+
Affix-Porter	=	=	=	=
Pre-Suf	-	-	-	-
Porter 2	+	+	+	+
Paice/Husk	=	=	=	+
Porter	+	+	+	+
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.16: Desc - non difficili+Difficili Significatività

- S:
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 8.16% per il MAP, 7.63% per l'R-precision, del 7.76% nell'F-score e vengono reperiti 305.0 documenti rilevanti in più rispetto alla baseline.
- Affix-Porter

  - Con questo algoritmo non vi sono differenze significative.

- Pre-Suf
  - Con questo algoritmo di stemming possiamo notare un decremento significativo in ogni misura analizzata. In particolare vediamo un peggioramento del 5.49% per il MAP, 3.98% per l'R-precision, del 2.19% nell'F-score e vengono reperiti 60.0 documenti rilevanti in meno rispetto alla baseline.

- Porter 2
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 10.27% per il MAP, 8.28% per l'R-precision, del 9.82% nell'F-score e vengono reperiti 398.0 documenti rilevanti in più rispetto alla baseline.

- Paice/Husk
  - Con questo algoritmo di stemming notiamo un miglioramento nel numero di documenti rilevanti recuperati, in particolare vengono reperiti 200.0 documenti rilevanti in più.

- Porter
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 9.29% per il MAP, 8.54% per l'R-precision, del 9.88% nell'F-score e vengono reperiti 398.0 documenti rilevanti in più rispetto alla baseline.

- Lovins
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

- Misto
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

Per le query basate solo sulla descrizione, l'algoritmo S, Porter e Porter 2 si distinguono, ottenendo miglioramenti significativi in tutte le misure, l'algoritmo di Paice/Husk invece porta un incremento significativamente solo nel numero di documenti rilevanti reperiti.

Indice	MAP	R-Precision	F1-Score (beta=1)	Num. Doc. Rilevanti Recuperati
S	0.2305	0.4332	0.2151	4609.0
Affix-Porter	0.2178	0.4120	0.2079	4479.0
Pre-Suf	0.1988	0.3860	0.1959	4209.0
Porter 2	0.2326	0.4383	0.2192	4726.0
Paice/Husk	0.2159	0.4185	0.2102	4494.0
No stemming	0.2136	0.4061	0.2006	4272.0
Porter	0.2314	0.4359	0.2184	4701.0
Lovins	0.2179	0.4115	0.2060	4415.0
Misto	0.2084	0.4044	0.2025	4342.0

Tabella 4.17: Title+Desc - non difficili+Difficili Risultati

Nome Indice	MAP	R-Precision	F1-Score	Num. Doc. Rilevanti Recuperati
S	+	+	+	+
Affix-Porter	=	=	=	+
Pre-Suf	-	-	=	=
Porter 2	+	+	+	+
Paice/Husk	=	+	+	+
Porter	+	+	+	+
Lovins	=	=	=	=
Misto	=	=	=	=

Tabella 4.18: Title+Desc - non difficili+Difficili Significatività

- S:
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 7.91% per il MAP, 6.68% per l'R-precision, del 7.21% nell'F-score e vengono reperiti 337.0 documenti rilevanti in più rispetto alla baseline.

- Affix-Porter
  - Con questo algoritmo vi è una differenza significativa solo nel numero di documenti recuperati. Per le altre misure non vi è una differenza significativa. In particolare notiamo che vengono reperiti 207.0 documenti rilevanti in più.
- Pre-Suf
  - Con questo algoritmo di stemming possiamo notare un decremento significativo nell'R-Precision e nel MAP. In particolare notiamo un decremento del 6.90% nel MAP e del 4.95% nell'R-Precision
- Porter 2
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 8.89% per il MAP, 7.94% per l'R-precision, del 9.28% nell'F-score e vengono reperiti 454.0 documenti rilevanti in più rispetto alla baseline.
- Paice/Husk
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata, ad eccezzione che per il MAP. In particolare vediamo un miglioramento del 3.07% per l'R-precision, del 4.79% nell'F-score e vengono reperiti 222.0 documenti rilevanti in più rispetto alla baseline.
- Porter
  - Con questo algoritmo di stemming possiamo notare un miglioramento significativo in ogni misura analizzata. In particolare vediamo un miglioramento del 8.34% per il MAP, 7.37% per l'R-precision, del 8.87% nell'F-score e vengono reperiti 429.0 documenti rilevanti in più rispetto alla baseline.
- Lovins
  - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

- Misto
    - Con questo algoritmo non si notano differenze significative rispetto alla baseline.

Quando si utilizzano sia la descrizione che il titolo come parte della query, gli algoritmi S, Porter 2, Paice/Husk, Affix-Porter e Porter mostrano miglioramenti significativi in almeno una delle misure per entrambi i tipi di topic. Anche in questo caso si nota come la lunghezza della query influsce sulla l'impatto che gli algoritmi di stemming hanno sull'efficacia del reperimento.

4.3 Discussione dei risultati

Come detto e visto in precedenza, alcuni algoritmi di stemming hanno un impatto sull'efficacia del reperimento, sia in termini positivi che negativi. Inoltre abbiamo notato che questo impatto varia in base sia alla lunghezza della query che per la difficoltà dei topic su cui si effettuano le interrogazioni. In particolare abbiamo visto che gli algoritmi di stemming risultano non avere un impatto significativo o, se presente è negativo, nei topic considerati difficili. Nei topic considerati non difficili invece alcuni algoritmi portano un incremento significativo.
Ponendoci in un contesto più realistico, cioè quando non si effettuano interrogazioni solo su topic difficili o non, si nota che alcuni degli algoritmi considerati hanno un impatto significativo. In particolare si nota che vi sono tre algoritmi che risultano impattanti nel effiacia del reperimento, questi tre algoritmi sono:

- S
- Porter
- Porter 2

Di questi tre algoritmi, quello che risulta avere un impatto positivo maggiore è l'algoritmo di Porter 2, l'algoritmo di Porter invece ottiene risultati molto simili, ma leggermente inferiori. L'S stemmer risulta anch'esso avere un impatto significativo positivo, ma si colloca come ultimo. Come detto in precedenza, il grado di miglioramento di questi algoritmi varia in base alla lunghezza della query. Si nota come la metrica MAP decresce in base alla lunghezza della query, in quanto raggiunge l'incremento significativo massimo nelle query corte. Le altre metriche invece ottengono un incremento percentuale maggiore nelle query medie, seguite poi dalle query lunghe ed infine dalle query corte.

Capitolo 5

Conclusioni

Nella tesi presentata, si è voluto analizzare se e quali algoritmi di stemming avessero un impatto sulle performance di un sistema di reperimento. Gli algoritmi selezionati su cui si è scelto di effettuare un confronto sono i seguenti:

- S
- Affix-Porter
- Pre-Suf
- Porter
- Porter 2
- Paice/Husk
- Lovins
- Misto

Inizialmente, partendo dalla collezione TREC Robust 2004, è stato costituito un vocabolario con le parole uniche presenti all'interno dei documenti della collezione al fine di calcolare su di esso una serie di metriche su cui trarre considerazioni riguardanti la forza degli algoritmi di stemming selezionati. La collezione TREC Robust 2004 è famosa per contenere delle esigenze informative difficili per i comuni motori di reperimento. Da questa

collezione sono stati estratti 9 insiemi di interrogazioni, ottenuti dalla combinazione della difficoltà dei topic (non difficili, difficili e tutti i topic) con la lunghezza delle interrogazioni (corte, medie e lunghe). Le query corte sono state estratte dal campo "title," le query medie dal campo "description," mentre le query lunghe sono state ottenute da una combinazione dei campi citati precedentemente. Per determinare se l'utilizzo di un algoritmo di stemming risultasse impattante nell'efficacia, sono state calcolate diverse misure per valutare se vi fosse una variazione significativa tra le baseline e le query in cui veniva utilizzato un algoritmo di stemming. Inizialmente, è stato effettuato un confronto tra le query della stessa lunghezza, ma effettuate su topic con diversi livelli di difficoltà. Dall'analisi dei risultati, emerge che gli algoritmi di stemming non hanno un impatto significativo positivo sui topic difficili per ogni lunghezza delle query. Tuttavia, si osserva un impatto positivo sulle query effettuate su topic "non difficili." In particolare, gli unici algoritmi che hanno un impatto positivo in tutte le metriche e per tutte le lunghezze analizzate risultano essere:

- S
- Porter
- Porter 2

Nelle query lunghe, a differenza delle query corte e medie, si osserva un impatto positivo sull'efficacia anche da parte dell'Affix-Porter, del Paice/Husk e dell'algoritmo di Lovins. Passando a un contesto più realistico, ovvero quando vengono effettuate query su topic sia non difficili che difficili, i risultati mostrano come gli algoritmi S, Porter e Porter 2 risultano significativamente impattanti in ogni metrica analizzata. Si osserva che il miglioramento del Mean Average Precision (MAP) raggiunge il massimo nelle query corte, mentre le altre misure presentano il massimo miglioramento nelle query medie. In generale, si nota quindi che l'efficacia di un algoritmo di stemming può variare in base alla lunghezza della query e alla "difficoltà" della query in cui viene applicato. Oltre a quanto detto in precedenza, però, l'efficacia di un algoritmo di stemming nel reperimento dell'informazione può variare, anche, in base ad altri fattori, tra cui i documenti della collezione, le query effettuate [2] e la lingua utilizzata [16]. Ad esempio, in una collezione di documenti tecnici o accademici, l'uso di una terminologia specifica può rendere meno efficace un algoritmo di stemming generico. Allo stesso modo, la natura delle query effettuate può influenzare l'efficacia

di un algoritmo di stemming, sempre per via della terminologia utilizzata. Per quanto riguarda la lingua utilizzata, nell'inglese Harman [9] ha osservato che vi sono risultati contrastanti (anche se successivamente Hull [17] e Krovetz [18] hanno osservato dei miglioramenti consistenti), per le altre lingue invece, come lo sloveno [19], l'olandese [20] e l'arabo [21] si è osservato un miglioramento significativo nelle performance relative al reperimento dell'informazione.

Bibliografia

[1] In: (). URL: https://it.wikipedia.org/wiki/Flessione_(linguistica).

[2] Jiaul H. Paik e Swapan K. Parui. "A Fast Corpus-Based Stemmer". In: *ACM Transactions on Asian Language Information Processing* 10.2 (2011). ISSN: 1530-0226. DOI: 10.1145/1967293.1967295. URL: https://doi.org/10.1145/1967293.1967295.

[3] J. Xu e W.B Croft. "Corpus-Based Stemming Using Cooccurrence of Word Variants". In: *ACM Transactions on Information Systems* (1998). DOI: 267954.267957. URL: https://doi.org/10.1145/267954.267957.

[4] Cristian Moral et al. "A survey of stemming algorithms in information retrieval". In: *Information Research* 19 (mar. 2014).

[5] M.F. Porter. "An algorithm for suffix stripping". In: *Program* 14.3 (1980), pp. 130–137. URL: https://tartarus.org/martin/PorterStemmer/def.txt.

[6] M.F. Porter. "Developing the English stemmer". In: (2002). URL: http://snowball.tartarus.org/algorithms/english/stemmer.html.

[7] Julie Beth Lovins. "Development of a Stemming Algorithm". In: *Information Storage and Retrieval* 4.3 (1968), pp. 107–132. URL: https://aclanthology.org/www.mt-archive.info/MT-1968-Lovins.pdf.

[8] Chris D. Paice. "Another Stemmer". In: *ACM SIGIR ForumVolume 24Issue 3Fall* (1990), pp. 56–61. DOI: 101306.101310.

[9] Donna Harman. "How Effective Is Suffixing?" In: *National Library of Medicine,* (1991).

[10] Diego Mollá e Ben Hutchinson. “Intrinsic versus Extrinsic Evaluations of Parsing Systems”. In: (apr. 2003), pp. 43–50. URL: https://aclanthology.org/W03-2806.

[11] Karen Sparck Jones e Julia R. Galliers. “Evaluating Natural Language Processing Systems: An Analysis and Review”. In: (1996). DOI: 10.5555/547445.

[12] NIST. “TREC 2004 Robust Track Guidelines”. In: (2004). URL: https://trec.nist.gov/data/robust/04.guidelines.html.

[13] William B. Frakes. “Strength and Similarity of Affix Removal Stemming Algorithms”. In: *Computer Science Department Virginia Tech* (2003).

[14] W. Bruce Croft, Donald Metzler e Trevor Strohman. “Search Engines - Information Retrieval in Practice”. In: (2009). URL: https://api.semanticscholar.org/CorpusID:2350758.

[15] Voorhees E.M. “Variantions in relevance judgments and the measurement of retrieval effectiveness”. In: *Information Processing and Management* (2000).

[16] Felipe N. Flores e Viviane P. Moreira. “Assessing the impact of Stemming Accuracy on Information Retrieval – A multilingual perspective”. In: *Information Processing Management* 52.5 (2016), pp. 840–854. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2016.03.004. URL: https://www.sciencedirect.com/science/article/pii/S0306457316300358.

[17] Hull. “Stemming algorithms - A case study for detailed evaluation”. In: (1996).

[18] R. Krovetz. “Viewing morphology as an inference process”. In: (1993).

[19] M. Popovic e Willet. “The effectiveness of stemming for natural-language access to Slovene textual data”. In: (1999).

[20] Kraaij e Pohlmann. “Viewing stemming as recall enhancement”. In: (1996).

[21] de Roeck e Al-Fares. “A morphologically sensitive clustering algorithm for identifying arabic roots”. In: (2000).

Appendice A

Tabelle

A.1 Algoritmo di Porter

Struttura algorimo di Porter.

- Step 1
  - Step 1a

SSES	->	SS
IES	->	I
SS	->	SS
S	->	-

  - Step 1b

(m > 0)	EED	->	EE
(*v*)	ED	->	-
(*v*)	ING	->	-

    Se la seconda o la terza regola vengono applicate, verranno applicate anche le seguenti regole:

-	AT	->	ATE
-	BL	->	BLE
-	IZ	->	IZE
(*d e non (*L, *S o *Z))	*d	->	lettera singola
(m = 1 e *o)		->	E

– Step 1c

(*v*)	Y	->	I

- Step 2

(m > 0)	ATIONAL	->	ATE
(m > 0)	TIONAL	->	TION
(m > 0)	ENCI	->	ENCE
(m > 0)	ANCI	->	ANCE
(m > 0)	IZER	->	IZE
(m > 0)	ABLI	->	ABLE
(m > 0)	ALLI	->	AL
(m > 0)	ENTLI	->	ENT
(m > 0)	ELI	->	E
(m > 0)	OUSLI	->	OUS
(m > 0)	IZATION	->	IZE
(m > 0)	ATION	->	ATE
(m > 0)	ATOR	->	ATE
(m > 0)	ALISM	->	AL
(m > 0)	IVENESS	->	IVE
(m > 0)	FULNESS	->	FUL
(m > 0)	OUSNESS	->	OUS
(m > 0)	ALITI	->	AL
(m > 0)	IVITI	->	IVE
(m > 0)	BILITI	->	BLE

- Step 3

(m > 0)	ICATE	->	IC
(m > 0)	ATIVE	->	-
(m > 0)	ALIZE	->	AL
(m > 0)	ICITI	->	IC
(m > 0)	ICAL	->	IC
(m > 0)	FUL	->	-
(m > 0)	NESS	->	-

- Step 4

(m > 1)	AL	->	-
(m > 1)	ANCE	->	-
(m > 1)	ENCE	->	-
(m > 1)	ER	->	-
(m > 1)	IC	->	-
(m > 1)	ABLE	->	-
(m > 1)	IBLE	->	-
(m > 1)	ANT	->	-
(m > 1)	EMENT	->	-
(m > 1)	MENT	->	-
(m > 1)	ENT	->	-
(m > 1 e (*S o *T)	IOT	->	-
(m > 1)	OU	->	-
(m > 1)	ISM	->	-
(m > 1)	ATE	->	-
(m > 1)	ITI	->	-
(m > 1)	OUS	->	-
(m > 1)	IVE	->	-
(m > 1)	IZE	->	-

Con questa regola si presuppone che i suffissi siano stati rimossi, le regole che seguiranno serviranno per correggere e sistemare eventuali errori

- Step 5
  - Step 5a

(m > 1)	E	->	-
(m > 1 e *d e *L)	*d	->	lettera singola

  - Step 5b

(m > 1)	E	->	-
(m > 1 e *d e *L)	*d	->	lettera singola

Questo algoritmo cerca di non rimuovere i suffissi se la radice rimanente è troppo corta e non basandosi su delle basi linguistiche può produrre delle radici non valide.

A.2 Algoritmo di Porter 2

Struttura dell'algoritmo di Porter 2[1]:

Struttura algorimo di Porter 2. Step 0

(m > 0)	'	->	-
(m > 0)	'S	->	-
(m > 0)	'S'	->	-

- Step 1
  - Step 1a: Viene rimossa la condizione per la rimozione del prefisso "IES", e viene aggiunta la rimozione del suffisso "IES". Viene rimpiazziato da "I" se preceduta da almeno una lettera, altrimenti viene viene sostituito da "IE"
  - Step 1b:
    - Viene aggiunto il prefisso "EEDLY", che viene rimpiazzato da "EE" se appartenente alla regione R1[2]
    - Vengono aggiunti i prefissi "EDLY" e "INGLY". Vengono rimossi se preceduti da una parola contenente una vocale, dopodichè, se la parola termina in "AT","BL" o "IZ" si aggiunte "E". Se la parola termina con una doppia viene rimossa l'ultima lettera o se la parola risultasse troppo corta viene aggiunta un "E" (per esercio "hop" -> "hope")
  - Step 1c:
    - Il suffisso "Y" viene rimosso se preceduto da una consonante se questa non è la prima lettera della parola
- Step 2: Vengno aggiunti alcuni suffissi
  - "BLI" viene sostituito con "BLE"
  - "OGI" viene sostituito con "OG", se preceduto dalla lettera "l"
  - "FULLI" viene sostituito con "FUL"
  - "LI" viene rimosso se preceduto da una radice valida

[1] verranno riportati solo i cambiamenti rispetto al primo algoritmo di Porter

[2] R1 è la regione dopo la prima non-vowel (consonante) che segue una vocale, o è la regione nulla alla fine della parola se non c'è una tale non-vowel

- Step 3:
  - ”TIONAL” viene sostituito con ”TION”
  - ”ACTIONAL” viene sostituito con ”ATE”
  - ”ATIVE” viene eliminato se nella regione R2
- Step 4: invariato
- Step 5
  - Unificati gli step 5a e 5b, per il resto invariato

A.3 Algoritmo di Lovins

Le terminazioni presentate da Lovins sono ordinate in base alla loro lunghezza, da 11 caratteri a 1. Ogni terminazione è poi seguita dal codice della condizione corrispondente:

alistically B	arizability A	izationally B

Tabella A.1: Suffissi da undici lettere

antialness A	arisations A	arizations A	entialness A

Tabella A.2: Suffissi da dieci lettere

allically C	antaneous A	antiality A	arisation A
arization A	ationally B	ativeness A	eableness E
entations A	entiality A	entialize A	entiation A
ionalness A	istically A	itousness A	izability A
izational A			

Tabella A.3: Suffissi da nove lettere

ableness A	arizable A	entation A	entially A
eousness A	ibleness A	icalness A	ionalism A
ionality A	ionalize A	iousness A	izations A
lessness A			

Tabella A.4: Suffissi da otto lettere

ability A	aically A	alistic B	alities A
ariness E	aristic A	arizing A	ateness A
atingly A	ational B	atively A	ativism A
elihood E	encible A	entally A	entials A
entiate A	entness A	fulness A	ibility A
icalism A	icalist A	icality A	icalize A
ication G	icianry A	ination A	ingness A
ionally A	isation A	ishness A	istical A
iteness A	iveness A	ivistic A	ivities A
ization F	izement A	oidally A	ousness A

Tabella A.5: Suffissi da sette lettere

aceous A	acious B	action G	alness A
ancial A	ancies A	ancing B	ariser A
arized A	arizer A	atable A	ations B
atives A	eature Z	efully A	encies A
encing A	ential A	enting C	entist A
eously A	ialist A	iality A	ialize A
ically A	icance A	icians A	icists A
ifully A	ionals A	ionate D	ioning A
ionist A	iously A	istics A	izable E
lessly A	nesses A	oidism A	

Tabella A.6: Suffissi da sei lettere

acies A	acity A	aging B	aical A
alist A	alism B	ality A	alize A
allic BB	anced B	ances B	antic C
arial A	aries A	arily A	arity B
arize A	aroid A	ately A	ating I
ation B	ative A	ators A	atory A
ature E	early Y	ehood A	eless A
elity A	ement A	enced A	ences A
eness E	ening E	ental A	ented C
ently A	fully A	ially A	icant A
ician A	icide A	icism A	icist A
icity A	idine I	iedly A	ihood A
inate A	iness A	ingly B	inism J
inity CC	ional A	ioned A	ished A
istic A	ities A	itous A	ively A
ivity A	izers F	izing F	oid A
oides A	otide A	ously A	

Tabella A.7: Suffissi da cinque lettere

able A	ably A	ages B	ally B
ance B	ancy B	ants B	aric A
arly K	ated I	ates A	atic B
ator A	ealy Y	edly E	eful A
eity A	ence A	ency A	ened E
enly E	eous A	hood A	ials A
ians A	ible A	ibly A	ical A
ides L	iers A	iful A	ines M
ings N	ions B	ious A	isms B
ists A	itic H	ized F	izer F
less A	lily A	ness A	ogen A
ward A	wise A	ying B	yish A

Tabella A.8: Suffissi da quattro lettere

acy A	age B	aic A	als BB
ant B	ars O	ary F	ata A
ate A	eal Y	ear Y	ely E
ene E	ent C	ery E	ese A
ful A	ial A	ian A	ics A
ide L	ied A	ier A	ies P
ily A	ine M	ing N	ion Q
ish C	ism B	ist A	ite AA
ity A	ium A	ive A	ize F
oid A	one R	ous A	

Tabella A.9: Suffissi da tre lettere

ae A	al BB	ar X	as B
ed E	en F	es E	ia A
ic A	is A	ly B	on S
or T	um U	us V	yl R
s' A	's A		

Tabella A.10: Suffissi da due lettere

a A	e A	i A	o A
s W	y B		

Tabella A.11: Suffissi da una lettera

Ogni terminazione è poi associatata a una condizione. Le condizioni sono invece le seguenti(* sta per ogni lettera):

- **A**: Nessuna restrizione sull'estrazione dello stem.
- **B**: L'estrazione della radice richiede una lunghezza minima di 3 caratteri.
- **C**: L'estrazione della radice richiede una lunghezza minima di 4 caratteri.
- **D**: L'estrazione della radice richiede una lunghezza minima di 5 caratteri.
- **E**: Non rimuove la terminazione delle parole che terminano con "e".
- **F**: Lunghezza minima della radice = 3 e non rimuove la terminazione delle parole che terminano con "e".
- **G**: Lunghezza minima della radice = 3 e la terminazione viene rimossa solo dopo "f".
- **H**: la terminazione viene rimossa e solo dopo "t" o "ll".
- **I**: Non rimuove la terminazione delle parole che terminano con "o" o "e".
- **J**: Non rimuove la terminazione delle parole che terminano con "a" o "e".
- **K**: Lunghezza minima della radice = 3 e la terminazione viene rimossa solo dopo "l", "i" o "u*e".
- **L**: Non rimuove la terminazione delle parole che terminano con "u", "x" o "s", a meno che "s" segua "o".
- **M**: Non rimuove la terminazione delle parole che terminano con "a", "c", "e" o "m".
- **N**: Lunghezza minima della radice = 4 quando la parola termina con "s**", altrimenti = 3.
- **O**: La terminazione viene rimossa solo dopo "l" o "i".

- **P**: Non rimuove la terminazione delle parole che terminano con ”c”.
- **Q**: Lunghezza minima della radice = 3 e non rimuove la terminazione delle parole che terminano con ”l” o ”n”.
- **R**: La terminazione viene rimossa solo dopo ”n” o ”r”.
- **S**: Rimuove la terminazione solo dopo ”dr” o ”t”, a meno che ”t” segua ”t”.
- **T**: Rimuove la terminazione solo dopo ”s” o ”t”, a meno che ”t” segua ”o”.
- **U**: Rimuove la terminazione solo dopo ”l”, ”m”, ”n” o ”r”.
- **V**: Rimuove la terminazione solo dopo ”c”.
- **W**: Non rimuove la terminazione delle parole che terminano con ”s” o ”u”.
- **X**: La terminazione viene rimossa solo dopo ”l”, ”i” o ”u*e”.
- **Y**: La terminazione viene rimossa solo dopo ”in”.
- **Z**: Non rimuove la terminazione delle parole che terminano con ”f”.
- **AA**: La terminazione viene rimossa solo dopo ”d”, ”f”, ”ph”, ”th”, ”l”, ”er”, ”or”, ”es” o ”t”
- **BB**: Lunghezza minima della radice = 3 e non rimuove la terminazione delle parole che terminano con ”met” o ”ryst”.
- **CC**: La terminazione viene rimosso solo dopo ”l”

Le regole di trasformazione gestiscono casistiche come le lettere doppie finale, i plurali irregolari o le stranezze morfologiche inglesi causate dal comportamento dei verbi latini nella seconda cogniugazione(assume / assumption, commit / commission).

1 Le lettere doppie: b, d, g, l, m, n, p, r, s, t vengono ridotte a una singola lettera.

2 ”iev” viene sostituito con ”ief”.

3 ”uct” viene sostituito con ”uc”.

4 ”umpt” viene sostituito con ”um”.

5 ”rpt” viene sostituito con ”rb”.

6 ”urs” viene sostituito con ”ur”.

7 ”istr” viene sostituito con ”ister”

7a ”metr” viene sostituito con ”meter”.

8 ”olv” viene sostituito con ”olut”

9 ”ul” viene sostituito con ”l” tranne nel caso in cui segue a, o, i.

10 ”bex” viene sostituito con ”bic”.

11 ”dex” viene sostituito con ”dic”.

12 ”pex” viene sostituito con ”pic”.

13 ”tex” viene sostituito con ”tic”.

14 ”ax” viene sostituito con ”ac”.

15 ”ex” viene sostituito con ”ec”.

16 ”ix” viene sostituito con ”ic”.

17 ”lux” viene sostituito con ”luc”.

18 ”uad” viene sostituito con ”uas”.

19 ”vad” viene sostituito con ”vas”.

20 ”cid” viene sostituito con ”cis”.

21 ”lid” viene sostituito con ”lis”.

22 ”erid” viene sostituito con ”eris”.

23 ”pand” viene sostituito con ”pans”.

24 ”end” viene sostituito con ”ens” tranne nel caso in cui segue ”s”.

25 ”ond” viene sostituito con ”ons”.

26 ”lud” viene sostituito con ”lus”.

27 ”rud” viene sostituito con ”rus”.

28 ”her” viene sostituito con ”hes” tranne nel caso in cui segue ”p” o ”t”.

29 ”mit” viene sostituito con ”mis”.

30 ”ent” viene sostituito con ”ens” tranne nel caso in cui segue ”m”.

31 ”ert” viene sostituito con ”ers”.

32 ”et” viene sostituito con ”es” tranne nel caso in cui segue ”n”.

33 ”yt” viene sostituito con ”ys”.

34 ”yz” viene sostituito con ”ys”

Sebbene siamo descritte come applicate in successione, vengono suddivise in due fasi:

- La prima regola viene eseguita nello step 1
- Una o nessuna delle restanti regole possono essere applicate nello step 2

A.4 Algoritmo di Paice/Husk

Regola	Sostituzione
ai*2.	-ia > - if intact
a*1.	-a > - if intact
bb1.	-bb > -b
city3s.	-ytic > -ys
ci2>	-ic > -
cn1t>	-nc > -nt
dd1.	-dd > -d
dei3y>	-ied > -y
deec2ss.	-ceed > -cess
dee1.	-eed > -ee
de2>	-ed > -
dooh4>	-hood > -
e1>	-e > -
feil1v.	-lief > -liev
fi2>	-if > -
gni3>	-ing > -
gai3y.	-iag > -y
ga2>	-ag > -
gg1.	-gg > -g
ht*2.	-th > - if intact
hsiug5ct.	-guish > -ct
hsi3>	-ish > -
i*1.	-i > - if intact
i1y>	-i > -y

Regola	Sostituzione
ji1d.	-ij > -id – see nois4j> & vis3j>
juf1s.	-fuj > -fus
ju1d.	-uj > -ud
jo1d.	-oj > -od
jeh1r.	-hej > -her
jrev1t.	-verj > -vert
jsim2t.	-misj > -mit
jn1d.	-nj > -nd
j1s.	-j > -s
lbaifi6.	-ifiabl > -
lbai4y.	-iabl > -y
lba3>	-abl > -
lbi3.	-ibl > -
lib2l>	-bil > -bl
lc1.	-cl > c
lufi4y.	-iful > -y
luf3>	-ful > -
lu2.	-ul > -
lai3>	-ial > -
lau3>	-ual > -
la2>	-al > -
ll1.	-ll > -l
mui3.	-ium > -
mu*2.	-um > - if intact
msi3>	-ism > -
mm1.	-mm > -m
nois4j>	-sion > -j
noix4ct.	-xion > -ct
noi3>	-ion > -
nai3>	-ian > -
na2>	-an > -
nee0.	protect -een
ne2>	-en > -

Regola	Sostituzione
nn1.	-nn > -n
pihs4>	-ship > -
pp1.	-pp > -p
re2>	-er > -
rae0.	protect -ear
ra2.	-ar > -
ro2>	-or > -
ru2>	-ur > -
rr1.	-rr > -r
rt1>	-tr > -t
rei3y>	-ier > -y
sei3y>	-ies > -y
sis2.	-sis > -s
si2>	-is > -
ssen4>	-ness > -
ss0.	protect -ss
suo3>	-ous > -
su*2.	-us > - if intact
s*1>	-s > - if intact
s0.	-s > -s
tacilp4y.	-plicat > -ply
ta2>	-at > -
tnem4>	-ment > -
tne3>	-ent > -

Regola	Sostituzione
tna3>	-ant > -
tpir2b.	-ript > -rib
tpro2b.	-orpt > -orb
tcud1.	-duct > -duc
tpmus2.	-sumpt > -sum
tpec2iv.	-cept > -ceiv
tulo2v.	-olut > -olv
tsis0.	protect -sist
tsi3>	-ist > -
tt1.	-tt > -t
uqi3.	-iqu > -
ugo1.	-ogu > -og
vis3j>	-siv > -j
vie0.	protect -eiv
vi2>	-iv > -
ylb1>	-bly > -bl
yli3y>	-ily > -y
ylp0.	protect -ply
yl2>	-ly > -
ygo1.	-ogy > -og
yhp1.	-phy > -ph
ymo1.	-omy > -om
ypo1.	-opy > -op
yti3>	-ity > -
yte3>	-ety > -
ytl2.	-lty > -l
yrtsi5.	-istry > -
yra3>	-ary > -
yro3>	-ory > -
yfi3.	-ify > -
ycn2t>	-ncy > -nt
yca3>	-acy > -
zi2>	-iz > -
zy1s.	-yz > -ys
end0.	