POLITECNICO DI TORINO

Collegio di ingegneria informatica,
del cinema e meccatronica

Corso di Laurea Magistrale in
Ingegneria informatica (Computer Engineering)

Tesi di Laurea Magistrale

Generazione automatica di riassunti di collezioni di documenti multilingua

Studio di tecniche basate su estrazione di itemset frequenti e
Latent Semantic Analysis

Relatori:
prof. Paolo Garza
prof. Luca Cagliero

Candidato:
Adriano Tittarelli

Anno accademico 2016-2017

Negli ultimi anni, l'utilizzo sempre più diffuso di internet ha fatto sì che il numero di documenti digitali prodotti e archiviati sia aumentato considerevolmente, rendendo sempre più indispensabili tecniche per l'estrazione automatica di informazioni. Nel presente lavoro viene presentato un metodo per la generazione automatica di riassunti di documenti testuali, in particolare di riassunti di tipo generico (non *query-based*), multidocumento, estrattivi e *sentence-based*. Il metodo si basa su tecniche di *latent semantic analysis* integrandole con l'estrazione di *itemset* frequenti e *n-gram*.

Indice

Capitolo 1

Introduzione

Negli ultimi anni si è assistito a una crescita imponente della disponibilità di informazioni digitali. La globalizzazione ha inoltre accentuato l'interesse verso informazioni provenienti da ogni parte del mondo e conseguentemente verso strumenti applicabili a diverse lingue.
In questa ottica, l'utilizzo di riassunti automatici, ottenuti con metodi *language-agnostic*, può essere un valido strumento nella ricerca e nella selezione dei documenti di interesse.
Nel tempo sono state sviluppate varie metodologie per la sommarizzazione che si basano su diversi principi e trovano applicazione in diversi contesti. Lo scopo è in ogni caso la produzione di un riassunto, di una lunghezza che può dipendere dal suo utilizzo, creato a partire da un documento o da un insieme di documenti.
I riassunti possono essere categorizzati in varie tipologie, che tengono conto sia di fattori tecnici nel metodo di generazione sia di scelte dettate dagli ambiti di applicazione. Molti dei sistemi esistenti utilizzano tecniche estrattive, generano cioè il riassunto selezionando parti del documento originale come frasi o paragrafi. Tecniche di tipo *abstractive* sfruttano invece meccanismi diversi, che implicano la generazione di linguaggio.
Per quanto riguarda la tipologia dei dati in ingresso, la necessità di estrarre informazioni dal web ha dato la spinta ad un modello di sommarizzazione multi-documento, in cui viene ricavato un unico riassunto a partire da una collezione di documenti relativi allo stesso argomento.
La sommarizzazione automatica trova vari ambiti di applicazione. Può essere utile ad esempio per determinare l'argomento su cui verte un testo ma anche per estrarre un riassunto finalizzato ad uno scopo specifico, la cui qualità è misurabile attraverso l'efficacia con cui il compito viene svolto utilizzando il surrogato dei documenti originali.
Il primo studio relativo alla sommarizzazione automatica di testi risale al 1958 e propone un metodo di tipo estrattivo basato sulla frequenza dei termini e il loro posizionamento reciproco. Studi successivi hanno ripreso l'uso della frequenza con modelli più complessi come la TF*IDF (*Term Frequency - Inverse Document Frequency*) o LLR (*Log Likelihood Ratio*). La ricerca in questo ambito è molto attiva e sono stati sviluppati metodi basati su

tecniche diversificate come lo studio della struttura del documento, le 'catene semantiche' e il *clustering*.
In alcuni casi si utilizzano tecniche mutuate da altri contesti come l'utilizzo degli *itemset* frequenti, tipici del *data mining*, o dei grafi. Alcuni dei metodi più interessanti sviluppati negli scorsi anni sono basati sulla *Latent Semantic Analysis* (LSA), che estrae e rappresenta il significato nel contesto d'uso di una parola, attraverso calcoli statistici applicati a un'ampia collezione di testi. Il metodo, sviluppato nel 1997 dalla *computer scientist* Susan Dumais e da Thomas Landauer, uno psicologo specializzato negli studi sull'interazione uomo-macchina, si basa sull'intuizione dei principi di funzionamento dell'apprendimento linguistico umano.
La LSA è basata sulla *Singular Value Decomposition (SVD)*, una tecnica di fattorizzazione delle matrici che, in termini intuitivi, riordina in modo semantico i contenuti della matrice originale.

L'obiettivo di questo lavoro è la creazione di un nuovo metodo di sommarizzazione estrattiva *multi-document* e in larga parte indipendente dalla lingua oggetto di analisi. L'implementazione è stata effettuata partendo dalla base dei metodi esistenti basati su LSA, ampliandoli tramite l'utilizzo di *itemset* frequenti come elemento di base del documento. L'idea è quella che gli *itemset* frequenti, che rappresentano combinazioni di parole, siano in grado di 'catturare' più informazioni di quanto non facciano i singoli termini.

Capitolo 2

Tecniche per la generazione automatica di riassunti

2.1 Formulazione del problema

Negli ultimi anni si è assistito a una crescita imponente della disponibilità di informazioni digitali. Da un lato con lo sviluppo del web e dei social network che ha portato alla produzione di contenuti da parte di un'ampia base di utenti[1], dall'altro con la forte tendenza alla digitalizzazione da parte di aziende ed enti pubblici e privati. La globalizzazione fa inoltre sì che l'interesse verso le informazioni acquisite sia spesso transnazionale. I tipi di informazioni raccolte variano da forme più strutturate e mappabili nei tradizionali database relazionali, a formati di concezione più recente e meno strutturati, che hanno portato allo sviluppo dei database NoSQL, nei quali la struttura dei dati non è predefinita, fino ad arrivare a informazioni testuali completamente destrutturate, i cui contenuti non sono accompagnati da alcun tipo di metadato. In questa sovrabbondanza di informazioni eterogenee è spesso indispensabile la creazione di metodi che rendano possibile l'analisi e la selezione di quantità di documenti che non sarebbero altrimenti vagliabili tramite una lettura puntuale.

In questa ottica l'utilizzo di riassunti automatici può essere un valido strumento nella ricerca e nella selezione dei documenti di interesse.

Nel tempo sono state sviluppate varie metodologie per la *summarization* che si basano su diversi principi e trovano applicazione in diversi contesti. Lo scopo è in ogni caso la produzione di un riassunto, di una lunghezza che può dipendere dal suo utilizzo, creato a partire da un documento o da un insieme di documenti.

[1] Nelle sua statistiche ufficiali Facebook dichiara di avere 2.01 miliardi di utenti attivi mensili alla data del 30 giugno 2017: `https://newsroom.fb.com/company-info/`

I riassunti possono essere categorizzati approssimativamente nelle seguenti tipologie, che tengono conto sia di fattori tecnici nel metodo di generazione sia di scelte dettate dagli ambiti di applicazione:

Extractive - abstractive Un riassunto di tipo *extractive* è composto da sottoparti del testo originale, che possono essere paragrafi, frasi o parte di esse. Queste sottoparti vengono selezionate o in base alla loro rilevanza generale nel documento o in base alla loro rilevanza verso un singolo argomento oggetto di ricerca da parte di un utente. Il metodo estrattivo si basa quindi sulla suddivisione del testo in sottoparti che devono poi essere scelte in base a un qualche criterio definito. Questo approccio è sicuramente quello più esplorato perché evita le complessità della generazione del linguaggio e perché permette l'utilizzo di algoritmi in larga parte indipendenti dalla lingua del testo. Tuttavia la sommarizzazione estrattiva presenta alcune forti limitazioni come ad esempio quelle evidenziate da Paice [48] [49], il quale pone il problema di come le frasi selezionate possano non essere autocontenute e portare riferimenti a frasi vicine ma non selezionate dal sistema estrattivo.
In un riassunto di tipo *abstractive*, che è la forma più consueta nella stesura di riassunti *human-generated*, le frasi vengono invece riformulate, compresse ed espresse eventualmente con termini differenti. Questo tipo di approccio può sicuramente dare potenzialmente risultati migliori ma presenta complessità maggiori legate alla generazione del linguaggio e al forte legame delle tecniche utilizzate con la lingua oggetto di analisi. Esistono anche delle forme particolari delle tipologie succitate che si adattano a scopi specifici; ad esempio la *headline summarization* può essere vista come un sottoinsieme delle precedenti in cui il riassunto consta di una sola frase. Infine, per alcune attività, in aggiunta o in alternativa a un riassunto tradizionale, può essere utile produrre un insieme di parole chiave rappresentative del documento e che potrebbero in teoria non apparire nel documento stesso.

Single document - multi document Nella sommarizzazione *single document* l'obiettivo è la produzione di un riassunto rappresentativo di un solo documento. Questa tecnica però non riesce ad adattarsi bene a tutti i contesti e ad alcuni nuovi schemi emersi recentemente. Con la crescita del web sono nati nuovi paradigmi di estrazione delle informazioni, spesso basati sul recupero massivo di documenti. Questo ha portato alla necessità di una sommarizzazione *multi document*. Questo diverso approccio si adatta infatti meglio alla ridondanza di informazioni presente in rete. L'obiettivo è quello di riassumere il contenuto di parecchie fonti raggruppate per argomento (una notizia, una ricerca o altro) in un unico documento breve che elimini le ridondanze, sia rappresentativo del gruppo di documenti originali e consenta di risalire eventualmente alle fonti. Questo approccio non comporta solo l'unione dei documenti in un unico testo finale ma deve anche gestire la presenza di eventuale ridondanza nei testi presi in considerazione.
La sommarizzazione di più documenti non necessariamente deve esplorare solo un topic

predeterminato. Dal gruppo di documenti potrebbero infatti emergere altre tematiche di rilievo, portando eventualmente alla luce informazioni interessanti di cui non si conosceva a priori l'esistenza.

Indicative - informative Questo tipo di suddivisione è basata sullo scopo per cui il riassunto è generato e, in maniera correlata, sulle sue dimensioni.
Un riassunto indicativo consente al lettore di capire gli argomenti di cui tratta il documento originale riportandone i passaggi più rilevanti. Un esempio sono i riassunti generati dai motori di ricerca. La lunghezza di un riassunto indicativo è tipicamente il 5-10% del testo originale [23].
Un riassunto informativo consta di un documento che può essere letto alternativamente al documento originale [43], conservandone i dettagli più importanti ma riducendone sostanzialmente le dimensioni. Un riassunto informativo può essere tipicamente lungo il 20-30% del testo originale [23].
Naturalmente questo tipo di suddivisione è indicativa e possono esistere variazioni atte a soddisfare esigenze specifiche.

Generic - query focused - update Un riassunto generico è generato senza conoscere a priori il tipo di utilizzo che se ne deve fare e deve essere rappresentativo al massimo del documento.
Un riassunto *query based* cerca invece le parti del documento che siano attinenti alla ricerca effettuata dall'utente, tralasciando o dando meno importanza ad altre parti, anche potenzialmente significative, ma non attinenti alla ricerca effettuata.
Un *update summary* è un riassunto che, data una collezione in cui nuovi documenti sono aggiunti periodicamente, mette in evidenza quelli che sono i nuovi contenuti introdotti dal documento più recente [59].

La varietà vista nei tipi di riassunto è dovuta al fatto che nessun metodo può dirsi migliore a priori. La metodologia scelta deve essere ottimale per il contesto e per il tipo di risultato che si intende raggiungere con la creazione del riassunto [24].

2.2 Esempi di contesti applicativi

La produzione di riassunti può avere ampi settori di applicazione. In generale può essere utile in tutti quei contesti in cui l'informazione persa analizzando il riassunto piuttosto che il documento originale, è compensata dal vantaggio ottenuto in termini di tempi e costi operazionali.

I riassunti creati da operatori comportano dei costi notevoli; un riassunto automatico, il cui costo al contrario può considerarsi marginale, dimostra un valore aggiunto nel

momento in cui permette di effettuare un'operazione in maniera più veloce e senza una perdita sensibile di accuratezza rispetto all'uso dei documenti integrali.

Sono stati effettuati numerosi studi per stabilire se i sistemi automatici di sommarizzazione siano in effetti efficaci in vari ambiti [43]. Ad esempio è stato dimostrato che i riassunti *query-based* consentono ad un operatore di stabilire la rilevanza di un documento per un argomento in maniera più veloce e con meno errori dovuti alla difficoltà di gestione della sovrabbondanza di informazioni, rispetto alla lettura dei documenti originali, mantenendo lo stesso livello di accuratezza [56]. Nello studio TIPSTER Text Summarization Evaluation (SUMMAC) [36] è stata analizzata l'efficacia nel lavoro di un analista di *intelligence*, comparando la capacità di stabilire se un documento fosse o meno rilevante per un certo argomento da un sommario o dal documento originale. Riassunti di circa il 17% del testo originale raddoppiavano la velocità nel prendere una decisione senza una perdita di accuratezza significativa.

Le potenzialità della sommarizzazione possono essere sfruttate anche in applicazioni in cui l'utilità è meno evidente. Il GMAT (Graduate Management Admissions Test) è un test per valutare l'attitudine di uno studente verso gli studi in materie economiche. Nel 2000 il test era somministrato da ETS (Educational Testing Service) che per la correzione di una delle parti del test, il tema, si avvaleva di un sistema automatico detto *e-rater* in grado supportare il valutatore assegnando dei punteggi ad alcuni aspetti della scrittura, come l'organizzazione delle idee, la varietà e la correttezza dei termini usati. In uno studio si è dimostrato che l'utilizzo di tecniche di sommarizzazione sui temi dei candidati portava a punteggi più alti in *e-rater* rispetto ai temi originali [8]. Gli autori concludono che il dato sia imputabile alla forte presenza di ridondanza nei temi, scritti sotto vincoli di tempo pressanti che non ne consentono un editing accurato. La versione riassunta può essere quindi usata sia per modellare meglio le funzionalità del valutatore automatico sia per dare al candidato un'utile informazione sulle criticità del proprio tema con la comparazione dei due testi.

In uno studio del 2001, Simone Tufel ha analizzato l'utilità della sommarizzazione automatica di articoli scientifici per un compito specifico, ovvero la valutazione del grado di relazione di un articolo con la ricerca esistente, esplicitando ad esempio quali articoli criticano e quali corroborano una certa teoria o su quali ricerche è basato l'articolo corrente [54]. Il metodo sviluppato si basa sull'utilizzo del *machine learning* per mettere in relazione caratteristiche identificabili del testo come la posizione e il numero delle citazioni, catalogate da etichette per identificarne la natura in fase di training e utilizzando un classificatore Naive Bayesiano [26].
I risultati mostrano che, per individuare quali degli approcci menzionati in un articolo sono criticati e quali invece supportati o estesi, l'uso di sommari automatici col metodo presentato è efficace quasi quanto un riassunto manuale ad hoc e più dell'abstract originale dell'articolo stesso.

La *multi-document summarization* si è rivelata utile nel compito di estrarre più informazioni possibile dai risultati di un motore di ricerca. Roussinov e Chen [50] hanno creato un'interfaccia di visualizzazione per motori di ricerca esistenti che presenta all'utente, invece dei singoli risultati, un riassunto dei documenti restituiti clusterizzati, concludendo che il sistema migliorava la capacità degli utenti di reperire informazioni.

Un utente potrebbe essere interessato nell'individuare, in un gruppo di documenti simili, uno o più aspetti relativi a un particolare argomento (attività che ricade sotto il nome di *instance-retrieval* o *aspectual retrieval*). Nel lavoro di Maña-López et al. [35] agli utenti veniva assegnato il compito di individuare più aspetti possibili per un argomento analizzando i risultati di un motore di ricerca. L'analisi di un riassunto ha dimostrato di essere più rapida e altrettanto efficace rispetto alla lettura del contenuto dei singoli risultati della ricerca.

In un altro studio [38] agli utenti veniva assegnato il compito di scrivere un report su un argomento specifico in un tempo limitato, facendo utilizzare a vari gruppi diverse fonti tra cui gli articoli originali e i riassunti di Newsblaster, un portale creato dalla Columbia University che implementa e combina varie delle tecnologie rivolte al linguaggio, come ad esempio *clustering*, categorizzazione del testo e sommarizzazione, per generare dei sommari automatici delle notizie online [37]. Utilizzando i riassunti di Newsblaster dei documenti clusterizzati gli utenti tendevano a generare report migliori riportando inoltre una maggiore soddisfazione per aver avuto un tempo maggiore per completare l'attività.

In generale l'uso di riassunti automatici può ottimizzare lo svolgimento di molte attività. Lo sforzo per migliorare la qualità dei riassunti prodotti rende questa tecnica efficace per un numero sempre più ampio di attività.

2.3 Stato dell'arte

2.3.1 Tecniche basate sulla frequenza dei termini

Il primo studio sulla creazione automatica di riassunti è stato condotto nel 1958 ad opera di H. Luhn ed era incentrato sulla sommarizzazione di articoli scientifici e di riviste [33]. L'idea di base, che ha influenzato molti degli studi successivi, era che alcune parole in un documento sono descrittive del suo contenuto e che le frasi che maggiormente rappresentano il contenuto sono quelle in cui queste parole si trovano vicine tra loro. Per determinare l'importanza dei singoli termini venivano ritenute più significative le parole con una frequenza maggiore, imponendo un limite oltre il quale l'importanza decresce (per evitare di annoverare tra le più significative parole di uso comune nel contesto in esame, come ad esempio la parola *legge* in un testo di giurisprudenza), stabilendo in maniera empirica il limite inferiore e superiore di frequenza che determinano il crescere e decrescere della curva. In pratica l'importanza di una frase è determinata sia dalla presenza di parole

rilevanti in base alla loro frequenza sia dalla vicinanza tra queste parole. Un altro concetto introdotto in questo studio e ancora altamente attuale, è quello per cui alcune parole di uso comune (es. congiunzioni, articoli, preposizioni o verbi molto utilizzati come essere o avere), non portano alcun valore nel determinare l'argomento del testo, per cui possono essere escluse a priori dall'analisi utilizzando una lista predefinita (*stop word list*) [43].

Studi successivi hanno utilizzato tecniche più complesse ma i metodi estrattivi in larga parte si basano ancora sull'idea di assegnare un indice di rilevanza a sottoparti del testo, di solito frasi o paragrafi, per poi selezionarle e unirle come riassunto.

In un altro studio [26] viene sviluppata l'idea che una migliore comprensione del documento può essere ottenuta studiando non solo il documento stesso ma una collezione di documenti in cui esso è inserito. In questa direzione, un metodo largamente utilizzato per stabilire la significatività di un termine è la TF*IDF (Term Frequency, Inverse Document Frequency), il cui indice aumenta con la frequenza del termine nel documento ma diminuisce con una funzione inversa del numero di documenti in cui il termine appare [51]. In questa formulazione la TF*IDF non va utilizzata su un singolo documento ma necessita di un corpus. Se un termine appare frequentemente in un documento e poco in altri è un buon indice di significatività di quel termine per quel documento. La TF*IDF di un termine è un buon indicatore di importanza di un termine ed è semplice da calcolare. Queste proprietà spiegano perché la TD*IDF è attualmente una delle caratteristiche più largamente utilizzate nella sommarizzazione estrattiva ed è presente in varie forme nella maggior parte dei sistemi attuali [43].

Un diverso metodo basato sulla frequenza dei termini è quello ideato da Dunning [12] che parte dall'assunto che la TF*IDF possa non rilevare la significatività di parole che siano particolarmente distintive di un topic rispetto ad altre e introduce l'utilizzo di una tecnica diversa detta LLR (*Log-likelihood ratio*) che permette di trovare nel testo quelle che verranno definite *topic signatures* in un successivo lavoro [30]. Le frasi che vengono selezionate sono quelle che contengono il numero più elevato di *topic signatures*.

2.3.2 Tecniche basate sulla semantica

Un approccio completamente diverso è quello di tenere in conto la struttura del testo. Un esempio in questo senso sono le *lexical chains* (catene lessicali) [7]. Il metodo è basato ampiamente su WordNet [40] un dizionario di sinonimi e contrari con indicazione delle relazioni tra i termini di tipo parte-tutto e generico-specifico, e sul suo utilizzo tramite algoritmi euristici. Una catena individua una sequenza di parole, anche non contigue, che abbiano una relazione nel dizionario utilizzato.

L'utilizzo di catene di parole ne permette la disambiguazione. Associando più parole, eventualmente ognuna con più significati si può risalire all'argomento che le unisce.

Ogni catena ha una lunghezza che ne determina la forza, data dal numero di parole e dall'omogeneità della catena stessa.

Un limite di approcci basati sulla semantica come le *lexical chains* è quello della forte dipendenza delle tecniche utilizzate verso gli strumenti disponibili per la lingua in oggetto.

2.3.3 Tecniche basate sulla struttura del documento

Nel 1969 Edmunson ha sviluppato un metodo basato su un approccio diverso, ovvero sulle caratteristiche strutturali del documento [13]. Il suo sommarizzatore prende in considerazione aspetti quali parole indicative (*cue words*), parole che compaiono nel titolo e negli headers, e indicatori strutturali quali il posizionamento delle frasi. Questo tipo di approccio ha dato risultati molto positivi ed è stato ripreso da metodi successivi anche in abbinamento ad altre tecniche.

2.3.4 Tecniche basate su *clustering*

La sommarizzazione multi document è basata sull'analisi di più documenti sullo stesso argomento. Le tecniche basate sul *clustering* sfruttano il fatto che in questa configurazione è molto probabile che ci siano più frasi, provenienti eventualmente da documenti diversi, che contengono informazioni analoghe. Frasi simili possono essere raggruppate insieme in *cluster*. I *cluster* con il più elevato numero di frasi rappresenteranno gli argomenti più rilevanti per la collezione e il riassunto può essere creato tramite la selezione di una frase rappresentativa da ognuno dei *cluster* più rilevanti. Questa tecnica viene adottata dal metodo SIMFINDER [22]. Altri studi hanno approfondito il metodo con l'utilizzo del *clustering* gerarchico incrementale che costruisce delle gerarchie per i cluster che vengono aggiornate all'aggiunta di ogni frase [59].

2.3.5 Tecniche basate su grafi

Un limite delle tecniche basate sul *clustering* è il fatto che una frase possa essere inserita in un solo cluster. I metodi basati sui grafi permettono in questo senso una maggiore flessibilità [43]. Metodi inizialmente sviluppati per ricavare informazioni sulla struttura del Web come HITS [25] e PageRank di Google [41] sono stati rielaborati nell'ambito della sommarizzazione, nella quale i nodi rappresentano le frasi e gli archi rappresentano il grado di similarità fra le frasi. Esempi di questo approccio sono i metodi LexRank [14], TextRank [39] e altri [58] [55] [61]. Nel 2013 Baralis et al. hanno presentato un metodo basato sui grafi detto GraphSum (Graph-based Summarizer) che per stabilire le correlazioni tra termini utilizza la ricerca di regole di associazione tipica del *data mining*.

2.3.6 Tecniche basate su *Latent Semantic Analysis*

La *Latent Semantic Analysis (LSA)* è una teoria e un metodo per estrarre e rappresentare il significato nel contesto d'uso di una parola attraverso calcoli statistici applicati a un'ampia collezione di testi [27]. Il metodo, sviluppato nel 1997 dalla computer scientist Susan Dumais e da Thomas Landauer, uno psicologo specializzato nello studio dell'interazione uomo-macchina, si basa sull'intuizione dei principi di funzionamento dell'apprendimento linguistico umano. L'idea alla base è che l'insieme di tutti i contesti di parole, in cui una parola appare o non appare, forniscano un insieme di vincoli reciproci che determina la similarità reciproca di parole e gruppi di parole [28]. In questo modo è possibile associare e raggruppare parole e documenti ad argomenti o pseudo-argomenti anche non esplicitamente citati. Il metodo prevede la rappresentazione di campioni di linguaggio, elaborati tramite la scomposizione in parole e in passaggi (frasi, paragrafi o documenti), in uno spazio semantico di elevate dimensioni. Le rappresentazioni tramite LSA hanno dimostrato la loro capacità di simulare un'ampia varietà di fenomeni cognitivi umani, che vanno dall'apprendimento dei termini, all'accostamento semantico di frasi e parole, alla comprensione del contesto [28]. La LSA è basata su un metodo matematico che deriva informazioni dai dati che riceve in input; è evidente che, rispetto ai meccanismi di apprendimento umano, il metodo non sia in grado di sfruttare tutta la parte di informazioni non strettamente legate ai termini utilizzati, come l'intonazione del linguaggio verbale, le esperienze sensoriali, il linguaggio del corpo e i contesti in cui avviene la comunicazione. Nonostante questo la LSA è un metodo che sperimentalmente è stato provato essere estremamente efficace nella riproduzione di alcuni aspetti cognitivi umani.

Lo strumento utilizzato per realizzare la LSA è la *Singular Value Decomposition (SVD)*. La SVD è una tecnica di fattorizzazione delle matrici alla cui ideazione hanno contribuito vari matematici tra cui Eugenio Beltrami [53]. È possibile effettuare la SVD su una qualsiasi matrice a elementi complessi. E' possibile considerare, senza perdere di generalità, una matrice A di dimensione $m \times n$, con $m \geq n$ a elementi reali. Questa matrice può essere scomposta come il prodotto di tre matrici:

$$A = U\Sigma V^T \tag{2.1}$$

dove U è una matrice ortonormale per colonne di dimensione $m \times n$. $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ è una matrice $n \times n$ diagonale i cui elementi diagonali sono valori singolari non negativi ordinati in maniera discendente e V è una matrice $n \times n$ ortonormale[2].

[2]Considerando una matrice in cui le colonne identificano le frasi e le righe i termini, l'assunzione che $m \geq n$ è vera solo per documenti di dimensioni limitate come pagine web, pubblicazioni scientifiche o articoli di giornale. Se andassimo ad analizzare in questo modo ad esempio un libro, il numero di frasi supererebbe la varietà dei termini e le dimensioni delle matrici derivanti dalla scomposizione andrebbero riformulate.

In forma matriciale:

$$\underbrace{\begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} u_{1,1} & \dots & u_{1,n} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ u_{m,1} & \dots & u_{m,n} \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} v_{1,1} & v_{2,1} & \dots & v_{n,1} \\ v_{1,2} & v_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ v_{1,n} & \dots & \dots & v_{n,n} \end{bmatrix}}_{V^T} \tag{2.2}$$

Ognuna delle colonne di U è detta vettore singolare sinistro e ognuna delle colonne di V è chiamata vettore singolare destro. Se $rango(A) = r$, allora per Σ è vero:

$$\sigma_1 \geq \sigma_2 \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0 \tag{2.3}$$

In termini intuitivi la LSA rappresenta il significato di una parola come una sorta di media del significato che assume nelle frasi in cui appare, e il significato di una frase è una sorta di media dei significati delle parole che contiene [28].

Uno dei principi fondamentali del metodo è la riduzione della dimensionalità del problema, un approccio per cui alcune informazioni possono essere estratte con più precisione, eliminando le dimensioni meno significative del modello che si utilizza per la rappresentazione. Nella SVD la riduzione della dimensionalità si ottiene considerando solo alcuni dei valori singolari, i primi k, e solo le prime k colonne di U e V. La matrice che si ottiene moltiplicando queste sottomatrici può considerarsi un'approssimazione della matrice A originale.

L'equazione 2.4 rappresenta le matrici della SVD troncata in cui A' è un'approssimazione di A di rango ridotto k.

$$\underbrace{\begin{bmatrix} a'_{1,1} & \dots & \dots & a'_{1,n} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ a'_{m,1} & \dots & \dots & a'_{m,n} \end{bmatrix}}_{A'} = \underbrace{\begin{bmatrix} u'_{1,1} & \dots & u'_{1,k} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ u'_{m,1} & \dots & u'_{m,k} \end{bmatrix}}_{U'} \underbrace{\begin{bmatrix} \sigma'_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma'_k \end{bmatrix}}_{\Sigma'} \underbrace{\begin{bmatrix} v'_{1,1} & v'_{2,1} & \dots & v'_{n,1} \\ \vdots & & & \vdots \\ v'_{1,k} & \dots & \dots & v'_{n,k} \end{bmatrix}}_{V'^T} \tag{2.4}$$

Questo metodo può essere utilizzato ad esempio per comprimere le immagini. Considerando l'immagine come una matrice di valori, che indicano la luminosità del pixel, si

può utilizzare il metodo descritto in precedenza per rappresentarla con le tre sottomatrici. Già con pochi valori singolari è possibile intuire il contenuto dell'immagine e con qualche decina la differenza con l'immagine originale è sempre meno percettibile. Un esempio è visibile in figura 2.1. La 'quantità di informazione' conservata può essere misurata come la somma dei σ conservati rispetto alla somma totale.
La SVD è utilizzata anche in altri contesti come la *computer vision* e l'analisi dei dati climatici.

Una delle prime implementazioni della LSA per l'analisi dei testi cercava di determinare quali documenti di una collezione fossero i più rilevanti per una query e si basava sulla creazione di una matrice di termini per documenti [15]. In questo metodo ogni elemento della matrice rappresenta se, o quanto spesso, un termine appare in un documento. Le query vengono rappresentate tramite vettori formati dalla combinazione pesata dei termini. I documenti vengono scelti in base alla loro similarità con la *query*, calcolata con la *cosine similarity* la quale, dati due vettori $\vec{u}$ e $\vec{v}$, è calcolata come il coseno dell'angolo θ tra di essi ovvero:

$$\cos\theta = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|}$$

Il primo metodo che propone invece l'utilizzo della LSA per la sommarizzazione di documenti è quello proposto da Gong e Liu nel 2001 [21]. Il metodo parte dall'idea di costruire una matrice $A = [A_1 A_2 \dots A_n]$ in cui ogni colonna A_i rappresenta il vettore della frequenza pesata dei termini all'interno della frase i-esima per poi eseguire la SVD sulla matrice frasi / termini ottenuta.

La SDV individua la struttura semantica latente del documento rappresentato dalla matrice A. In questa operazione il documento originale viene scomposto in r vettori base linearmente indipendenti che rappresentano dei 'concetti'. Una caratteristica peculiare della SVD è la sua capacità di raggruppare semanticamente termini e frasi.

Ad esempio considerando le parole *dottore*, *medico*, *ospedale*, *medicina* e *infermiera*, i termini *dottore* e *medico* sono sinonimi nel contesto e *ospedale*, *medicina* e *infermiera* sono termini strettamente correlati. I sinonimi *dottore* e *medico* appaiono spesso in contesti simili, in cui appaiono spesso anche le parole *ospedale*, *medicina* e *infermiera*. Proprio per questi pattern ricorrenti le due parole saranno mappate vicine nello spazio singolare r-vettoriale.

Inoltre se una combinazione di parole è saliente e ricorrente in un documento, questo *pattern* sarà catturato e rappresentato da uno dei valori singolari. La grandezza del corrispondente valore singolare rappresenta l'importanza di questo *pattern* all'interno del documento.

Ogni frase contenente questa combinazione di termini sarà proiettata lungo il vettore singolare e la frase che meglio rappresenta il *pattern* avrà il valore di indice più alto per questo vettore. Visto che ogni combinazione di parole descrive un certo concetto nel documento è lecito ipotizzare che ogni vettore singolare rappresenti un concetto saliente del documento e che la misura del corrispondente valore singolare rappresenti l'importanza del topic nel documento [21].

Viste le suddette considerazioni gli autori propongono il seguente metodo per generare un riassunto di N frasi:

1. Suddividere il documento in frasi e impostare $k = 1$
2. Costruire la matrice frasi / termini
3. Effettuare la SVD sulla matrice per ottenere la matrice dei valori singolari Σ e la matrice dei vettori singolari destri V^T. Nello spazio dei vettori singolari ogni frase è rappresentata da $\psi_i = [v_{i1} v_{i2} \dots v_{ir}]^T$, vettore colonna di V^T
4. selezionare il k-esimo vettore singolare destro dalla matrice V^T
5. selezionare la frase associata all'indice del valore più grande del vettore singolare destro selezionato ed includerla nel sommario. In questa operazione si trova in pratica la frase più attinente al concetto individuato dal valore singolare k
6. Se si è raggiunto il numero di frasi preimpostato N fermarsi, altrimenti incrementare k di uno e tornare al passo 4.

Per la valutazione dei risultati, i sommari ottenuti vengono confrontati con dei riassunti generati manualmente da tre diversi operatori, con lo stesso principio di selezione delle frasi più rappresentative utilizzato dal sommarizzatore automatico, attraverso un indice che tenga conto sia della *recall* (un indice della completezza delle informazioni reperite rispetto a quelle presenti nei riassunti manuali) sia della *precision* (indice della stretta attinenza delle informazioni reperite sempre rispetto a quelle presenti nei riassunti manuali)[3]. I risultati riportati dagli autori sono paragonabili a quelli di altri metodi, corroborando le ipotesi considerate. Viene evidenziata anche la difficoltà nella valutazione della sommarizzazione generica, dimostrata dalla scarsa uniformità dei riassunti manuali, che cresce con il crescere della lunghezza del testo.

Murray, Renals e Carletta [42] hanno utilizzato il metodo precedente con una lieve variazione (selezionando più di una frase per ogni riga di V^T in numero proporzionale al valore del σ corrispondente) per la sommarizzazione di trascrizioni di parlato. Il parlato ha caratteristiche peculiari che influenzano l'efficacia dei sistemi di sommarizzazione.

[3] Una definizione più formale di *precision* e *recall* è riportata nel paragrafo 4.2.2

I risultati di questa forma di LSA vengono confrontati con un altro metodo basato sulla *Maximal Marginal Relevance* [9], una misura della dissimilarità tra la frase presa in considerazione e le frasi già selezionate per l'estrazione, i cui risultati sono paragonabili a quelli della LSA per questo ambito.

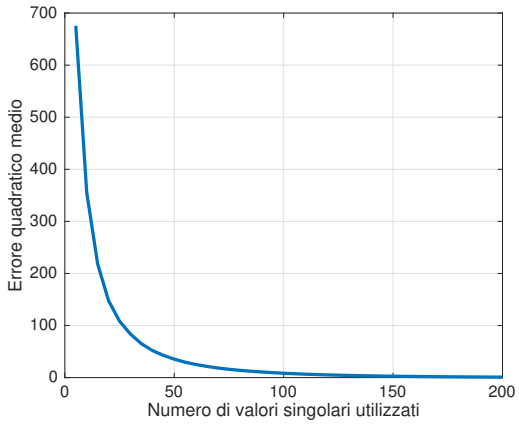
Un'altra diretta evoluzione del metodo di Gong e Liu è quello ideato da Steinberger e Ježek nel 2004 [52]. Gli autori partono dalla considerazione che il metodo di Gong e Liu estrae la frase più significativa per ognuno dei topic più rilevanti, individuati dagli autovalori più grandi. Con questo approccio si tralasciano però frasi potenzialmente indicative per più topic ma il cui indice non è il maggiore in assoluto in nessuno di essi. Utilizzano quindi lo stesso metodo per la generazione della matrice A, ma studiano una nuova metrica per la significatività delle frasi. Gli elementi della matrice V^T rappresentano il grado di importanza di ciascun topic per ciascuna frase. Il metodo proposto è quello di calcolare, per un numero di valori singolari fissato, una metrica della frase che tenga conto del peso del topic i-esimo σ_i e del peso del topic i-esimo per la k-esima frase $v_{k,i}$, tramite la seguente formula:

$$s_k = \sqrt{\sum_{i=1}^{n} (v_{k,i}^2 \cdot \sigma_i^2)} \tag{2.5}$$

dove s_k (salience score) è la lunghezza del vettore della frase k-esima nello spazio vettoriale modificato. Vengono selezionate le frasi con *salience score* maggiore e che sono quindi più rappresentative per i topic con valori singolari maggiori. In questo modo vengono selezionate frasi potenzialmente molto rappresentative del testo ma che con il metodo precedente sarebbero state scartate.

Una ulteriore variante nel calcolo del *salience score* è stata proposta da Ozsoy et al. [47]. Nel loro metodo, dopo il calcolo della SVD, vengono analizzate una alla volta le righe della matrice V^T calcolandone il valore medio, il quale rappresenta il valore medio di un concetto su tutte le frasi. Se un valore della riga è inferiore alla media viene azzerato nel tentativo di eliminare le correlazioni lasche tra frasi e *topic*. In seguito viene calcolato il *salience score* con il metodo di Steinberger e Ježek utilizzando la matrice V^T modificata.

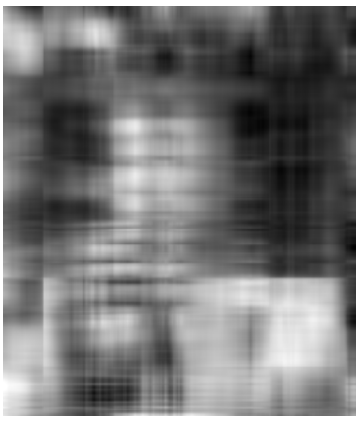
(a) Immagine originale



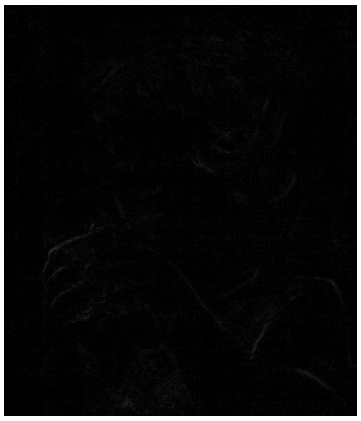
(b) Diagramma degli errori

(c) Immagine ricostruita utilizzando 65 valori singolari

(d) Immagine ricostruita utilizzando 20 valori singolari



(e) Immagine ricostruita utilizzando 5 valori singolari



(f) Errore nell'immagine con 65 valori singolari

(g) Errore nell'immagine con 20 valori singolari

(h) Errore nell'immagine con 5 valori singolari

Figura 2.1: Esempio di utilizzo della SVD per la compressione di immagini

2.3.7 Tecniche basate su *itemset* frequenti

La ricerca delle regole di associazione è un interessante metodo di *data mining* per cercare informazioni nei dati che siano non palesi e a volte diverse dalle aspettative. Uno dei primi studi in questo settore è stato condotto da Agrawal et al. nel 1993 [2]. L'ambito dello studio è quello delle transazioni in un supermercato.

Attraverso l'informatizzazione dei registratori di cassa, l'uso delle carte fedeltà e altri strumenti, le catene della GDO dispongono di una grande mole di dati riguardanti gli acquisti effettuati dai clienti. L'analisi di questi dati può rivelare informazioni fondamentali per una migliore programmazione delle vendite, per ottimizzare l'assortimento e i criteri di esposizione delle merci al fine di massimizzare i profitti.

Viene definita come base per l'analisi la transazione (*basket transaction*), un insieme di oggetti comprati insieme non necessariamente nello stesso momento ma eventualmente anche in un periodo di tempo. L'insieme delle informazioni raccolte sulle transazioni è noto come database transazionale. Un *itemset* è invece un sottoinsieme degli oggetti nella transazione. Una regola di associazione individua in *itemset* come antecedente e uno come conseguente. Ogni regola in un database ha una certa *confidence*. Ad esempio se individuiamo che nel 90% delle transazioni in cui vengono acquistati pasta e guanciale vengono acquistate anche le uova abbiamo l'itemset pasta, guanciale come antecedente e uova come conseguente: $\{pasta, guanciale\} \Rightarrow \{uova\}$ con una *confidence* del 90%.

Alla base degli algoritmi per l'individuazione delle regole di associazione c'è il sottoproblema dell'individuazione degli *itemset* frequenti, che può anche essere visto come un problema indipendente e la cui utilità va oltre la ricerca delle regole di associazione.

Un concetto importante in questo ambito è quello di supporto. Il supporto di un *itemset* indica in quante transazioni esso appare in rapporto al numero totale di transazioni. Un *itemset* frequente è un *itemset* con un supporto oltre una soglia prefissata. Un *itemset* frequente P si dice massimale se P non è incluso in nessun altro itemset frequente. Si dice invece chiuso se P non è incluso in nessun *itemset* che sia a sua volta incluso nella stessa transazione in cui è incluso P.

Un primo tentativo di utilizzo degli *itemset* frequenti per la sommarizzazione dei documenti è dovuto a Hynek e Ježek [23] in un lavoro del 2003 in cui si presentava un sistema di sommarizzazione *query based* formato da un sistema base comune e dei 'motori' intercambiabili per lo svolgimento dell'elaborazione principale. Uno di questi metodi proposti era basato sulla creazione preventiva di una tassonomia di argomenti e sull'utilizzo degli *itemset* frequenti per valutare l'aderenza di una frase o un paragrafo agli argomenti stessi.

Nel 2012 Baralis et al. [5] hanno presentato un nuovo metodo basato sull'utilizzo degli *itemset* frequenti. Il documento viene trasformato in un'insieme di transazioni, in cui ogni frase rappresenta una transazione e ogni transazione è composta dagli stem della frase

originaria. Dal database transazionale viene estratto un modello del documento che include gli *itemset* frequenti più informativi che non siano ridondanti utilizzando l'algoritmo proposto da Mampaey et al. nel 2011 [34]. La scelta delle frasi è effettuata tramite un *relevance score* che tenga in considerazione sia il valore di tf-idf associato ai singoli termini sia la rappresentatività da parte della frase del modello basato sugli itemset frequenti.

In un altro lavoro del 2015 Baralis et al. [6] presentano una nuova metodologia basata sull'utilizzo dei *weighted frequent itemset*, introdotti da Wang et al. nel 2000 [60]. Il metodo si distingue da quello citato precedentemente anche per l'utilizzo di una diversa misura della frequenza dei termini (tf-df) che risulta più adatta in una collezione di documenti omogenei.

Capitolo 3

LSA-itemset summarizer

3.1 Introduzione

L'obiettivo di questo lavoro è di formulare un metodo per la sommarizzazione automatica *multi-document* che sia estrattivo, generico (non *query-based*) e quasi completamente indipendente dalla lingua dei documenti analizzati.
L'implementazione è stata effettuata partendo dalla base dei metodi esistenti basati su LSA, in particolare quello di Steinberger e Ježek, ampliandoli tramite l'utilizzo di *itemset* frequenti come elemento di base del documento. L'idea è quella che gli *itemset* frequenti siano in grado di 'catturare' più informazioni di quanto non facciano i singoli *stem*. Sono state studiate anche altre variazioni che utilizzano *n-gram* di termini e la combinazione di *itemset* frequenti e *n-gram*.

Il metodo, denominato LSA-itemset summarizer o LSA-i, è stato progettato per ricevere in input uno o più documenti e generare un riassunto di tipo estrattivo *sentence-based*, la cui lunghezza sia un parametro del programma.

Nelle fasi di elaborazione iniziale vengono utilizzati degli strumenti di analisi del testo specifici per la lingua in oggetto, mentre i passi successivi sono completamente indipendenti dalla lingua.

Il metodo può essere schematizzato in quattro fasi principali descritte di seguito e rappresentate in figura 3.1:

1. **Elaborazione dei documenti e scomposizione in frasi.** In questa fase i documenti sono elaborati tramite una serie di filtri che prima suddividono il testo in frasi, poi eliminano i caratteri che non appartengono ai termini come tag, virgolette, parentesi, ecc., trasformano tutti i caratteri in minuscolo, rimuovono le *stopwords*, ovvero le parole prive di valore semantico significativo come ad esempio articoli, congiunzioni e verbi ausiliari, trasformano i termini in una forma base, comune a tutte le coniugazioni e declinazioni, detta *stem*. Il risultato è quindi una lista di frasi in un doppio formato (originale/sequenza di *stem*). Da questa lista vengono rimosse

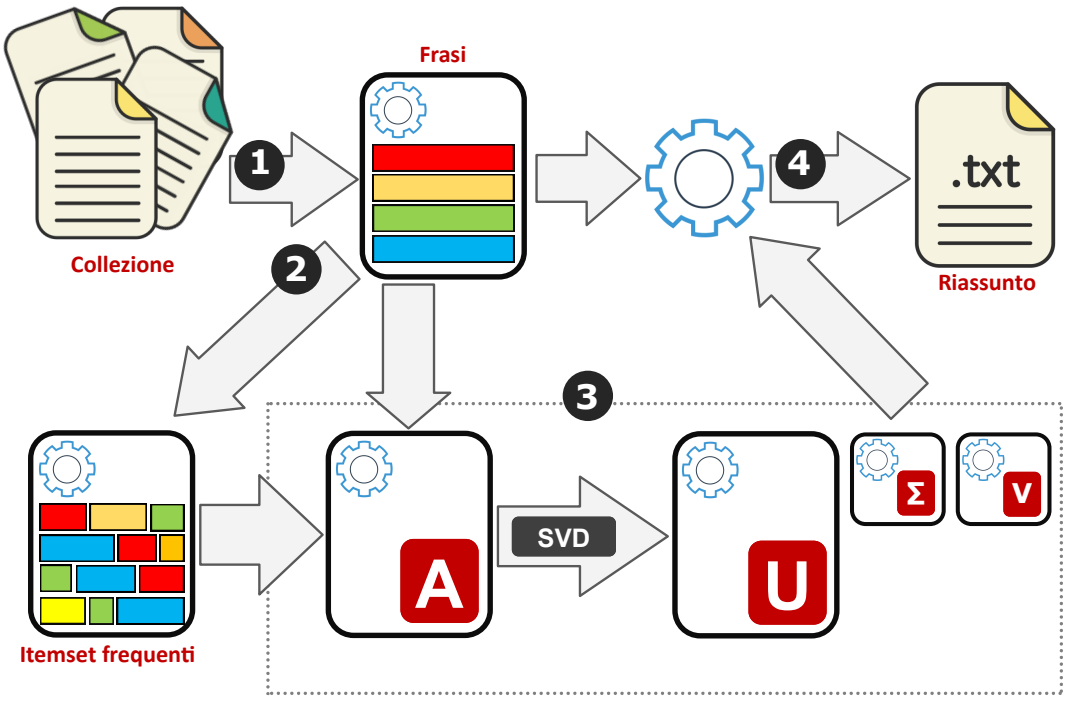


Figura 3.1: Blocchi funzionali del metodo

eventuali frasi che siano molto simili, eventualità possibile nella sommarizzazione *multi-document*.

2. **Generazione della lista di *itemset* frequenti.** A partire dagli *stem* contenuti nelle frasi analizzate viene estratta una lista di *itemset* frequenti, ovvero di insiemi di *stem* che ricorrono nelle frasi con una frequenza minima prefissata (supporto). Altre due varianti analizzate prevedono l'estrazione di una lista di *n-gram*, ovvero di sequenze di lunghezza limitata di *stem* consecutivi o di *itemset* frequenti calcolati a partire non dagli *stem* ma bensì dagli *n-gram*. Visto che le fasi di elaborazione successive sono indipendenti dalla variante utilizzata si farà riferimento genericamente agli '*itemset* frequenti'.

3. **Generazione della matrice frasi / *itemset* frequenti e decomposizione ai valori singolari**. Le informazioni generate nelle fasi precedenti vengono utilizzate per popolare una matrice le cui n colonne identificano le frasi e le cui m righe sono associate agli *itemset* frequenti. Gli elementi della matrice rappresentano il peso/legame tra coppie (*itemset* frequente, frase). Alla matrice viene applicata la decomposizione

ai valori singolari per identificare i legami principali tra frasi e itemset:

$$A_{m\times n} = U_{m\times n} \cdot \Sigma_{n\times n} \cdot V^T_{n\times n}$$

4. **Selezione delle frasi e generazione del riassunto** L'interpretazione semantica della scomposizione, derivata dalla LSA, è che gli elementi della matrice diagonale Σ, detti valori singolari, rappresentino i *topic* presenti nel testo e che gli elementi della matrice V^T rappresentino la rilevanza di una frase per un determinato topic. Per individuare le frasi da selezionare per il riassunto, viene calcolato un indice di rilevanza della frase (*salience score*). L'indice deve tenere conto sia della rilevanza del topic sia della rilevanza di una frase per quel topic specifico. Si è scelto quindi di adottare il metodo individuato da Steinberger e Ježek che calcola una sorta di 'lunghezza' della frase nello spazio vettoriale, per cui il *salience score* della k-esima frase diventa:

$$s_k = \sqrt{\sum_{i=1}^{n} (v_{k,i}^2 \cdot \sigma_i^2)}$$

con n uguale al numero di valori singolari che si intende prendere in considerazione.

3.2 Parametri

Il metodo proposto riceve in ingresso vari parametri che ne determinano il comportamento nelle diverse fasi di elaborazione.

Parametri relativi agli *itemset* frequenti Per l'individuazione dei *itemset* frequenti è stato utilizzato l'algoritmo LCM di Takeaki Uno [57] configurandolo con i seguenti parametri:

lcmExecutable. Sono state testate due diverse modalità dell'algoritmo: *frequent itemset* e *frequent closed itemset*. Nella prima modalità vengono selezionati tutti gli *itemset* con lunghezza e supporto sufficiente. Come visto nel paragrafo 2.3.7 i *frequent closed itemset* sono invece il sottoinsieme minimo dei precedenti che conservi le informazioni sul supporto. Il parametro indica quale delle due modalità utilizzare.

minSupport. Il supporto minimo indica quante volte un *itemset* deve apparire nel testo, in rapporto al numero complessivo di *itemset*, per essere preso in considerazione. Un supporto minimo uguale a zero equivale all'individuazione di tutti gli *itemset* del documento.

minItemsetLength. questo parametro rappresenta il numero minimo di *stem* di cui deve essere composto un *itemset*. Il valore uno equivale a selezionare tutti i possibili *itemset* che abbiamo un supporto adeguato indipendentemente dalla loro lunghezza.

Parametri generali:

language. Nel costruire questo metodo uno degli obiettivi è stato quello di renderlo il più possibile indipendente dalla lingua oggetto della sommarizzazione, tuttavia alcune delle fasi di elaborazione ne sono per loro natura dipendenti. La *stopword elimination* dipende da dizionari precostituiti diversi per ogni lingua. Lo *stemming* è fortemente legato alla lingua, delle cui peculiarità deve tenere conto, ed è inoltre inapplicabile nelle lingue in cui non esistono declinazioni e coniugazioni come ad esempio il mandarino. La fase di suddivisione in frasi, infine, è una problematica complessa che viene di solito affrontata dai software specializzati tramite la creazione di un modello addestrato per riconoscere l'inizio e la fine di una frase. Questi modelli vengono generati dando in input al software dei testi in cui la suddivisione delle frasi è annotata con un *markup* specifico. I modelli vengono in seguito utilizzati dal software per effettuare la suddivisione di testo non annotato. Il modello dà buoni risultati solo per la stessa lingua con cui è stato effettuato l'addestramento e risente anche del dominio del documento. Articoli di giornale e chat su internet ad esempio non necessariamente condividono le stesse modalità di suddivisione delle frasi.

similarityThreshold. Nelle collezioni di documenti utilizzate, e descritte più in dettaglio nel paragrafo 4.1, vengono predisposti per la sommarizzazione più articoli sul medesimo argomento. Per alcuni argomenti si ha la presenza di due frasi in due articoli diversi che hanno in larga parte lo stesso contenuto. Se una di queste frasi dovesse avere uno score elevato ed essere selezionata per il riassunto è probabile che anche l'altra lo sia, generando una inutile ridondanza nel riassunto. Si è deciso quindi di inserire un parametro che rappresenti la soglia di 'similarità' oltre la quale solo una delle due frasi viene inserita nella matrice per il calcolo della SVD.

matrixGenerationMethod. Per la generazione della lista corrispondente alle righe della matrice A sono state utilizzate tre tecniche diverse: *Frequent itemset*, *n-gram* o una combinazione dei due metodi precedenti in cui i *frequent itemset* non vengono generati a partire dagli *stem* bensì dagli *n-gram*.

maxMatrixSize. Dalla combinazione dei parametri precedenti dipende la dimensione della matrice A. Ad esempio valori elevati di *minSupport* e *minItemsetLength* potrebbero portare ad una matrice vuota o non rappresentativa. Allo stesso tempo valori troppo bassi potrebbero portare a dimensioni della matrice non gestibili in memoria.

3.3 Implementazione del metodo

Il metodo LSA-itemset summarizer è stato realizzato in Java. Per alcune operazioni sui testi sono state utilizzate delle librerie *open source* di larga diffusione come Apache

Lucene [3] e Apache OpenNLP [4]. Per le operazioni sulle matrici, in particolare per la scomposizione SVD è stata utilizzata la libreria EJML (Efficient Java Matrix Library) [1]

Il programma può essere configurato secondo i parametri visti in precedenza e riceve in input il percorso della cartella in cui sono contenuti i file di cui eseguire la sommarizzazione.

3.3.1 Elaborazione dei documenti e scomposizione in frasi

Stacking

Il programma scansiona la cartella ricevuta in *input* per individuare i file di testo. Ognuno del file letti viene inserito come elemento di una struttura in cui il percorso del file è la chiave e il contenuto è il valore.

Sentence detection

Una delle strutture dati più importanti nel funzionamento del programma è la frase. Si è scelto di rappresentarla tramite una classe che contenesse non solo la stringa estratta ma anche i formati con *stem* e *frequent itemset*.

Come prima fase è quindi necessario individuare quali siano le frasi che compongono i documenti.

Visto che il metodo si basa sulla selezione delle frasi che meglio riassumono il testo, la scomposizione del documento diventa di fondamentale importanza. Dalla qualità della scomposizione può dipendere sia la rappresentatività del riassunto sia la sua leggibilità.

Per l'hindi esiste un simbolo specifico per la terminazione di una frase detto *poorna viraam* e rappresentato nell'alfabeto devangari [1] con un tratto verticale detto *Devanagari Danda*, carattere Unicode U+0964: "।", questo rende la suddivisione in frasi realizzabile con l'uso di comuni funzioni di *text processing*.

Tutte le altre lingue in esame, comprese quelle non latine, per la terminazione delle frasi utilizzano il punto. Tuttavia lo stesso carattere è utilizzato per le abbreviazioni, nei numeri decimali, come separatore delle migliaia e per altri scopi. La semantica del carattere è dunque ambigua, a meno di conoscere la struttura della frase.

In alcune circostanze non è ovvio come la suddivisione debba essere effettuata per una persona, la questione è ancora più complessa per un calcolatore. Prendendo ad esempio questo testo:

> *"A destructive widespread tsunami threat does not exist based on historical earthquake and tsunami data," the US Pacific Tsunami Warning Center*

[1] il devangari è un alfabeto comune a varie lingue dell'India (sanscrito, hindi, marathi, kashmiri, sindhi, nepalese)

> *said. "However, there is the possibility of a local tsunami that could affect coasts located usually no more than a 100 km [60 miles] from the earthquake epicentre."*

è possibile effettuare la suddivisione in vari modi. Ad esempio l'ultimo virgolettato potrebbe costituire una frase a sé, che però tolta dal contesto potrebbe non essere un buon candidato per un riassunto leggibile.

Esistono alcuni software di *Natural Language Processing (NLP)* in grado di effettuare questa operazione. Questi software si basano su principi di *machine learning* e vanno 'addestrati' attraverso l'elaborazione di documenti annotati che devono essere simili a quelli che saranno poi soggetti di analisi. Per questo motivo ogni lingua richiede un training specifico del software. Le lingue per le quali è stato predisposto il software sono: arabo, ceco, inglese, francese, greco, ebraico e hindi, ovvero quelle utilizzate per il TAC2011 (vedi paragrafo 4.1.2). Per alcune delle lingue europee sono reperibili i modelli per Apache OpenNLP ma per altre non è stato possibile trovare dei modelli pronti da usare. Si è deciso quindi di realizzare delle funzioni ad hoc per il *sentence detection* implementando una serie di regole che, per quanto imperfetta, permettesse l'applicabilità del metodo. È indubbio che per ottimizzare le prestazioni sarebbe auspicabile preparare dei file di modello per un software di NLP; tuttavia le funzioni di *sentence detection* scritte hanno dato risultati di poco inferiori a quelli di Apache OpenNLP per le lingue per cui erano disponibili entrambe le opzioni.

Le operazioni compiute sul testo per suddividerlo in frasi sono mirate a sostituire tutte le occorrenze del carattere punto (.) che non rappresentano un delimitatore di frase, con un carattere speciale (§) per poi suddividere in frasi in corrispondenza dei caratteri 'punto' rimasti e infine ri-sostituire i caratteri speciali con un punto. L'algoritmo per le lingue diverse dall'hindi può quindi essere approssimato dall'algoritmo 3.1.

Tokenization

Una frase può contenere molti caratteri che non individuano una parola, come ad esempio la punteggiatura, le virgolette e le parentesi. Visto che i termini sono alla base del metodo, è necessaria una fase di elaborazione in cui vengono estratte e isolate le singole parole.

La distinzione tra maiuscolo e minuscolo potrebbe apportare significato in alcune circostanze, come nella distinzione tra nomi propri e sostantivo o aggettivi (es. ***Rossi*** e *rossi*). Tuttavia l'eventuale differenziazione potrebbe portare a considerare come diversi, termini che invece si distinguono solo per il fatto di trovarsi in una posizione diversa della frase, ad esempio dopo un punto. Sulla base di queste considerazioni si è utilizzato un filtro per la tokenizzazione che rendesse anche minuscoli i termini individuati. Il filtro, prima di applicare il minuscolo, spezza la frase ogni volta che individua un carattere che non sia una lettera o un numero. Per l'implementazione si è scelto di utilizzare a cascata il filtro

Algoritmo 3.1: Suddivisione del testo in frasi

```
/* Contenuto completo di un file                                    */
Input: s
/* elenco di abbreviazioni comuni nelle varie lingue                */
Data: abbreviations[]
/* Un array di stringhe, un elemento per ogni frase                 */
Output: sentences[]
foreach a  abbreviations[] do
    Sostituisci a. con a§ in s
end
/* Escape dei decimali / separatori di migliaia                     */
Sostituisci numero.numero con numero§numero in s
sentences ←suddividi s in corrispondenza del carattere "."
foreach sentence  sentences[] do
    Sostituisci § con . in sentence
end
return sentences[]
```

StandardFilter di Apache Lucene per spezzare in token, il LowercaseFilter per portare i caratteri in minuscolo e, solo per alcune lingue dove ha senso farlo, di utilizzare un filtro aggiuntivo detto ElisionFilter che rimuove alcune parole con elisioni[2] dallo scarso valore semantico.

La stringa:

"No more than a 100 km [60 miles] from the earthquake epicentre."

con i filtri suddetti produrrebbe la seguente lista di *token*:

[no, more, than, a, 100, km, 60, miles, from, the, earthquake, epicentre]

Stopword elimination

Come indicato nel paragrafo 2.3 una *stop word list* è una lista di parole utilizzate in maniera molto comune in una lingua, come congiunzioni, preposizioni e verbi di uso comune come essere o avere. La *stop word list* di Apache Lucene per la lingua italiana ad

[2] L'elisione è una forma di contrazione che si può avere in parole che terminano con vocali non accentate e che si trovino davanti a una parola che inizia con una vocale o con la lettera 'h'. Principalmente si utilizza per articoli e pronomi ma anche per altri morfemi. Alcuni esempi possono essere: *l'indice*, *d'Itaca*, *anch'io*

esempio è costituita da poco meno di trecento termini [3]. Riutilizzando l'esempio visto in precedenza la frase, dopo la rimozione delle stop words potrebbe avere una configurazione di questo tipo:

[no, 100, km, 60, miles, earthquake, epicentre]

Con la rimozione delle *stopwords* si ha quindi una riduzione della dimensione del problema affrontato e si evita di esaltare la rilevanza di parole che hanno scarso significato, in quegli algoritmi in cui è importante la cardinalità dei termini e la loro co-occorrenza.

Stemming

Molte lingue utilizzano versioni coniugate e declinate delle stesse parole. Nell'analisi di un testo può essere utile ad esempio individuare i termini 'nazione' e 'nazioni' come se fossero la stessa parola. Esistono delle tecniche, dette *stemming* e *lemmatizzazione* [4], per ricondurre le parole ad una forma base o canonica, comune tra tutte le sue coniugazioni o declinazioni.

Questo è un possibile output di uno *stemmer* sulla frase precedente:

[no, 100, km, 60, mile, earthquak, epicentr]

Per eseguire questi processi sono disponibili diversi algoritmi, sviluppati dagli anni sessanta in poi, per la lingua inglese (es. Porter, Lovins, Paice/Husk). Esiste anche il progetto Snowball [5] iniziato dal Dr. Martin Porter, che ha ideato un linguaggio di elaborazione delle stringhe attraverso il quale è possibile generare degli algoritmi di stemming a partire da una serie di regole per la rimozione dei suffissi.

In Apache Lucene esistono *stemmer* o *lemmatizer* per tutte le lingue utilizzate durante gli esperimenti.

Con l'operazione di *stemming* termina il processo di individuazione dei termini nel programma. Per tutte le operazioni successive gli *stem* individuati sono utilizzati come unità minima per l'elaborazione.

---

[3] `https://github.com/apache/lucene-solr/blob/master/lucene/analysis/common/src/resources/org/apache/lucene/analysis/snowball/italian_stop.txt`

[4] La lemmatizzazione è un processo più complesso rispetto allo *stemming*. Lo *stemming* opera con metodi euristici sulla parola così come è scritta riducendola ad una forma base. Così facendo si possono generare delle collisioni in cui termini di significato diverso vengono portati allo stesso stem. La lemmatizzazione invece cerca di individuare il corretto significato del termine in base al suo contesto prima di trasformarlo nella sua forma canonica

[5] `http://snowball.tartarus.org/texts/introduction.html`

Sentence pruning

In una collezione di documenti è possibile che ci siano frasi molto simili tra di loro o persino identiche. Ad esempio un *web crawler* [6] potrebbe recuperare articoli di giornale identici o molto simili, come nel caso di edizioni dello sesso articolo destinate a diversi paesi, come se fossero documenti indipendenti. Un esempio tratto da una collezione del TAC2011 (vedi paragrafo 4.1.2) è visibile in figura 3.2.

File A:
The sailors and marines, from the **Type 22** frigate HMS Cornwall, had been inspecting, in accordance with UN Security Council Resolution 1723, a ship that was believed to be smuggling cars into Iraq, though it was subsequently cleared after inspection.

File B:
The sailors and marines from the frigate HMS Cornwall had been inspecting, in accordance with UN Security Council Resolution 1723, a ship that was believed to be smuggling cars into Iraq (though it was subsequently cleared after inspection)**, when Iranian gunboats surrounded the sailors and arrested them at gunpoint.**

Figura 3.2: TAC2011 - Parti di due diversi documenti della stessa collezione molto simili tra loro. In neretto le differenze

La presenza di queste frasi duplicate apporta nuovo significato alla collezione in maniera marginale. È inoltre molto probabile che se una di queste frasi venisse scelta per il riassunto, la stessa cosa accadrebbe per la sua omologa. Per questo si è deciso di introdurre un parametro per eliminare le frasi simili secondo una soglia percentuale indicata dal parametro stesso. La 'similarità' tra due frasi è stata definita come la percentuale di *stem* della frase più lunga presente anche nella frase più corta.

Il dettaglio del procedimento è schematizzato nell'algoritmo 3.2.

Le frasi vengono confrontate mutualmente e, in caso di frasi con similarità sopra la soglia stabilita, viene mantenuta solo la frase con maggior numero di *stem*. Le frasi rimaste sono quindi inserite in una collezione che le associa univocamente a un numero progressivo che rappresenterà l'indice della frase nella matrice A. L'introduzione del *sentence pruning* è uno degli aspetti in cui LSA-itemset summarizer si differenzia dai precedenti algoritmi basati su LSA.

[6] Software che scansiona la rete cercando ed eventualmente salvando contenuti secondo criteri predefiniti come ad esempio articoli su un certo argomento

Algoritmo 3.2: Funzione che calcola se le frasi sono simili

```
/* Le due frasi nel formato con gli stem                    */
Input: s1, s2
/* Soglia percentuale di similarità                         */
Input: threshold
Function areSimilar (s1, s2, threshold)
    sLonger ←frase con più stem
    sShorter ←frase con meno stem
    matching ←numero di stem in comune tra le due frasi
    similarity ←matching / numeroStem(sLonger)
    if similarity > threshold then
        return true
    else
        return false
    end
```

3.3.2 Generazione della lista di *itemset* frequenti

Una volta individuate le frasi e i loro relativi *stem* si procede alla redazione di una lista di *stem* univoci con numerazione progressiva.
Nel metodo di Gong e Liu e in quello di Steinberger e Ježek, descritti nel paragrafo 2.3.6, gli indici degli *stem* rappresenterebbero l'indice dei termini nella matrice A, nella quale l'elemento $a_{i,j}$ indicherebbe la frequenza pesata del termine i nella frase j.

Questa fase, all'interno di LSA-itemset summarizer consiste nella creazione di una analoga lista di elementi che caratterizzeranno le righe della matrice A. Per l'implementazione sono stati previsti tre diversi metodi alternativi.

Generazione degli *itemset* frequenti

In questo approccio, in sostituzione degli *stem*, per la generazione della matrice A, vengono utilizzati gli *itemset* frequenti, intesi come gruppi di parole che appaiono nelle singole frasi e con una frequenza minima all'interno della collezione di documenti.

Per individuare gli *itemset* frequenti si utilizza l'algoritmo LCM di Takeaki Uno [57]. LCM prende in input un insieme di transazioni [7]. Una transazione è rappresentata da una lista di interi non duplicati [2]. Ogni frase viene quindi convertita in una lista di interi

[7] La terminologia del *data mining* è influenzata dal fatto che i primi studi sono stati effettuati sulle abitudini di acquisto di clienti di supermercati. Una transaction è quindi uno 'scontrino' di un cliente, contenente una serie di acquisti (*items*)

che rappresentano gli indici degli *stem* che la compongono, dalla quale vengono rimossi i duplicati.
Questo insieme di transazioni viene dato in input all'algoritmo LCM insieme al supporto minimo e alla lunghezza minima degli *itemset* che si desidera ricevere in output. Il risultato è una lista di *itemset* con a fianco il relativo supporto, ovvero il numero di volte che l'*itemset* è presente nell'insieme delle transazioni.
Gli *itemset* così individuati sono associati a un numero progressivo che ne rappresenterà l'indice sulle righe delle matrice A.

Generazione degli n-gram

Uno dei limiti riconosciuti della LSA è di non tenere in alcuna considerazione l'ordine in cui appaiono i termini. I pesi all'interno della matrice A dipendono esclusivamente dalla presenza o meno di un termine in una frase. Anche nel caso del metodo con *frequent itemset* l'importanza dell'ordine delle parole non viene presa in considerazione. Per cercare di recuperare queste informazioni è stata implementata una versione alternativa di LSA-itemset summarizer in cui la lista dei *frequent itemset* è sostituita da una lista di *n-gram* di lunghezza massima di quattro *stem* e ai quali vengono applicati gli stessi parametri utilizzati per i *frequent itemset* (lunghezza minima e supporto). Gli *n-gram* sono costituiti da sequenze di *stem* consecutivi e conservano le informazioni sull'ordine dei termini.

Approccio misto *n-gram / frequent itemset*

Come detto gli *n-gram* conservano le informazioni sull'ordine dei termini ma, essendo composti da termini consecutivi, non tengono in considerazione le sequenze all'interno delle quali si inseriscano sporadicamente altri termini, informazione che viene invece ben modellata dagli *itemset* frequenti. Nel tentativo di sfruttare gli aspetti positivi di entrambi gli approcci è stata modellata una terza variante nella quale si costruisce una lista di *n-gram* frequenti, lista che viene poi utilizzata come base per la generazione di *itemset* frequenti. In pratica si costruisce una lista di *itemset* frequenti di *n-gram* invece che di *stem*. La combinazione delle due tecniche può essere implementata in varie combinazioni, in particolare relativamente ai parametri da utilizzare nelle due fasi. La scelta effettuata è stata quella di utilizzare una configurazione fissa nella generazione degli *n-gram* (lunghezza minima 1, lunghezza massima 4, supporto 0) e di variare i parametri relativi all'estrazione degli *itemset* frequenti.

3.3.3 Generazione della matrice frasi / *itemset* frequenti e decomposizione ai valori singolari

Generazione della matrice frasi / *itemset* frequenti

La matrice A è una matrice sparsa in quanto non tutti gli *itemset* frequenti compaiono in tutte le frasi. Ipotizzando di avere n frasi e m *itemset* frequenti l'approccio seguito è quello di generare una matrice di zeri di dimensione $m \times n$ e valorizzare successivamente gli elementi dove si rileva la presenza di uno o più *itemset*.

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix} \tag{3.1}$$

L'elemento $a_{i,j}$ della matrice A sarà diverso da zero se l'*itemset* frequente i è presente nella frase j.

Ci sono varie modalità possibili per valorizzare gli elementi non nulli di A che tengono conto in maniera differente della distribuzione degli *itemset* frequenti. Nel loro studio Gong e Liu [21] analizzano varie di queste modalità (*weighting schema*). In particolare definiscono due diversi pesi, uno globale $G(i)$ che tiene conto della frequenza del termine nel documento e uno locale $L(i)$ che tiene conto della frequenza del termine all'interno della frase.

Il peso locale può essere determinato in base a quattro possibili alternative:

1. Non pesato: $L(i) = tf(i)$ dove $tf(i)$ è il numero di volte che il termine appare nella frase.

2. Binario: $L(i) = 1$ se il termine appare nella frase, $L(i) = 0$ se non appare.

3. Aumentato: $L(i) = 0.5 + 0.5 \cdot (tf(i)/tf(max))$ dove $tf(max)$ è la frequenza del termine con più occorrenze nella frase.

4. Logaritmico: $L(i) = log(1 + tf(i))$.

Il peso globale può valere:

1. Non pesato: $G(i) = 1$ per ogni termine

2. Frequenza inversa nel documento: $G(i) = \log(N/n(i))$ dove N è il numero totale di frasi presenti nel documento e $n(i)$ è il numero di frasi che contengono il termine i.

Il valore degli elementi della matrice è quindi determinato dall'equazione:

$$a_{i,j} = L(t_{i,j}) \cdot G(t_{i,j})$$

Dopo aver testato il loro algoritmo con le possibili combinazioni gli autori concludono che non ci siano differenze significative nell'utilizzo dei vari *weighting schema* nel loro metodo.
I concetti espressi riguardo i *weighting schema* da Gong e Liu e riferiti ai termini sono direttamente mappabili in LSA-itemset summarizer sugli *itemset* frequenti e partendo da questo presupposto nello sviluppo sono stati messi alla prova solo i metodi con $G(i) = 1$ e $L(i)$ non pesato o binario. Dopo alcuni test iniziali si è deciso di utilizzare lo schema binario. Un elemento $a_{i,j}$ della matrice A assume dunque il valore 1 se la frase j-esima contiene l'*itemset* i-esimo, il valore 0 in caso contrario.

La procedura per la creazione della matrice può essere riassunta dall'algoritmo 3.3

Algoritmo 3.3: Costruzione della matrice A frasi / *itemset* frequenti

```
/* Array di frasi e itemset frequenti                    */
Input: sentences[], frequentItemsets[]
m ← frequentItemsets.size()
n ← sentences.size()
/* matrice di zeri di dimensione m × n                   */
A[m][n] ← ∅_{m,n}
for j ← 1 to n do
    for i ← 1 to m do
        if (sentences[j] contiene frequentItemsets[i]) then
            A[i][j] ←1
    end
end
```

Scomposizione SVD della matrice

La matrice A viene scomposta tramite la SVD nel prodotto $U\Sigma V^T$. Per ricavare le tre matrici è stata utilizzata la libreria EJML (Efficient Java Matrix Library) [1] che lavora con matrici dense di *double*. Considerando di avere n frasi e m itemset frequenti si hanno le seguenti dimensioni delle matrici:

$$A_{m \times n} = U_{m \times n} \cdot \Sigma_{n \times n} \cdot V^T_{n \times n}$$

In Java un *double* occupa 8 byte, per una dimensione totale di

$$2(m \cdot n + n^2) \cdot 8 \; byte$$

In caso di dimensioni elevate, plausibili utilizzando a esempio un supporto minimo molto basso, l'occupazione in memoria può superare la disponibilità dell'elaboratore utilizzato. Nel corso dello sviluppo del software è stato rilevato empiricamente che documenti di lunghezza analoga possono generare matrici di dimensioni molto diverse in base alla varietà di *stem* individuati e alla loro co-occorrenza. Risulta quindi difficile stabilire a priori dei parametri che impediscano alle matrici di superare dei limiti prefissati.

È stato scelto quindi di procedere con un metodo per tentativi successivi che, in caso di superamento delle dimensioni massime della matrice impostate tramite parametro, ricominci dall'estrazione degli *itemset* frequenti ma con un supporto aumentato.

Come illustrato nel paragrafo successivo, per i fini della determinazione di un indice di rilevanza delle frasi, non è necessario l'output completo della SVD. Vengono utilizzati solamente i valori singolari più grandi e una porzione della matrice V^T. Esistono dei metodi di calcolo della SVD 'parziale' che restituiscono un numero limitato di valori singolari, una sottomatrice di U con le colonne di indice più basso e una sottomatrice di v^T con le righe di indice più basso. Questo tipo di calcolo può essere effettuato anche a partire da una matrice sparsa che ha una occupazione di memoria sensibilmente inferiore.
Ad esempio in Matlab è disponibile la funzione:

$$[U, \Sigma, V] = svds(A, k)$$

che restituisce i primi k valori singolari, le prime k colonne di U e le prime K colonne di V^T:

$$\underbrace{\begin{bmatrix} u_{1,1} & \dots & u_{1,k} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ u_{m,1} & \dots & u_{m,k} \end{bmatrix}}_{U} \quad \underbrace{\begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_k \end{bmatrix}}_{\Sigma} \quad \underbrace{\begin{bmatrix} v_{1,1} & v_{2,1} & \dots & v_{n,1} \\ v_{1,2} & v_{2,2} & & 0 \\ \vdots & & \ddots & \vdots \\ v_{1,k} & 0 & \dots & v_{n,k} \end{bmatrix}}_{V^T}$$

È evidente che, nei casi in cui il numero di *itemset* estratti sia elevato, questo tipo di rappresentazione potrebbe ridurre notevolmente l'occupazione di memoria da parte delle strutture dati.

Una ulteriore considerazione è necessaria per testi molto lunghi. Gli esperimenti svolti finora sono stati incentrati su collezioni di articoli di giornale. Il numero di frasi di una delle collezioni prese in considerazione è generalmente inferiore alle 300 unità. Il numero di *stem* è sempre inferiore alle 1500 unità mentre il numero di *itemset* frequenti può arrivare a superare le centinaia di migliaia nelle configurazioni con supporto molto basso.

L'analisi di testi molto lunghi, come ad esempio un libro, potrebbe richiedere risorse maggiori rispetto a quelle disponibili e il metodo e richiederebbe probabilmente una riformulazione di qualche tipo, come ad esempio una sommarizzazione parziale per unità logiche.

3.3.4 Selezione delle frasi e generazione del riassunto

Calcolo del salience score delle frasi

Una volta calcolata la SVD della matrice è possibile utilizzare le matrici Σ e V^T per il calcolo del *salience score*, ovvero per individuare quali sono le frasi più significative per il documento.

L'interpretazione 'semantica' del contenuto delle matrici ottenute dalla SVD, già formulata da Landauere e Dumais [27], è che i valori singolari, cioè i valori sulla diagonale della matrice Σ, rappresentino dei temi (topic) presenti all'interno del documento e che il loro valore sia un indice della rilevanza del topic per il documento stesso.

Per la creazione del riassunto possono essere presi in considerazione solo i valori singolari di dimensione maggiore. I criteri con cui scegliere quanti valori singolari prendere in considerazione possono essere diversi e dipendere anche dalla lunghezza desiderata per il riassunto. Il criterio che è stato seguito è quello di tagliare i valori singolari il cui valore sia inferiore alla metà del valore singolare maggiore. Visto che i valori singolari sono ordinati dalla SVD in maniera decrescente si utilizzano i σ_i per cui

$$\sigma_i \geqslant \frac{\sigma_0}{2}$$

Questo criterio può essere modificato, ad esempio se si desidera estrarre un riassunto di una determinata lunghezza o se è noto a priori quanti siano gli argomenti di rilevanza per il documento. Va fatto notare che la determinazione dei *topic* da parte della SVD è basata sulle co-occorrenze dei termini ma naturalmente non è sempre aderente ai reali *topic* trattati nel testo.

La matrice V^T contiene invece le informazioni che mettono in relazione le frasi con i topic. In pratica il valore di un elemento $v_{i,j}$ di V rappresenta l'*importanza* della frase i-esima per il topic j-esimo.

Per individuare le frasi da selezionare per il riassunto viene calcolato un indice di rilevanza della frase (*salience score*). L'indice di rilevanza deve tenere conto sia della rilevanza del topic sia della rilevanza di una frase per quel topic specifico. Si è scelto quindi di adottare il metodo individuato da Steinberger e Ježek che calcola una sorta di 'lunghezza' della frase nello spazio vettoriale, per cui il *salience score* della k-esima frase diventa:

$$s_k = \sqrt{\sum_{i=1}^{n} (v_{k,i}^2 \cdot \sigma_i^2)}$$

con n uguale al numero di valori singolari che si intende prendere in considerazione.

In appendice A è possibile visualizzare un esempio di come la SVD metta in relazione gli elementi della matrice.

Generazione del riassunto

Una volta associato ad ogni frase il relativo *salience score* il riassunto viene generato estraendo le frasi con l'indice più elevato. Il riassunto ha una maggiore leggibilità se le frasi vengono riportate non in ordine di *salience score* ma nell'ordine in cui si trovavano originariamente all'interno del testo.

3.4 Complessità

La maggior parte delle operazioni compiute nell'elaborazione dei testi per la generazione del riassunto ha dei tempi di calcolo e delle richieste di memoria limitate e una complessità che varia linearmente con la lunghezza complessiva della collezione di documenti analizzata. Per l'esecuzione dei due algoritmi più complessi, LCM e SVD, sono invece necessarie ulteriori considerazioni.

La complessità dell'algoritmo LCM è limitata da una funzione lineare del numero di *frequent closed itemset* [57], ma il loro numero cresce in maniera non lineare rispetto alla lunghezza della collezione in ingresso ed è dipendente dai contenuti del testo. Le dimensioni che il problema può assumere non inficiano la fattibilità dell'estrazione degli itemset frequenti ma si vanno a ripercuotere sull'esecuzione della SVD.

La complessità della SVD dipende dall'algoritmo utilizzato per implementarla. Uno dei migliori metodi a livello computazionale, R-SVD, impiega un tempo computazionale dell'ordine di $\mathcal{O}(4m^2n + 22n^3)$ [20]. I tempi di calcolo verificati sperimentalmente, rimangono nell'ordine di alcuni secondi per le matrici più grandi utilizzate, ma l'occupazione di memoria può facilmente superare la disponibilità dell'elaboratore con alcune combinazioni di parametri di configurazione.

Capitolo 4

Esperimenti

4.1 Dataset

Le funzionalità e le performance del software sono state valutate utilizzando, come input, delle collezioni documentali create per delle note conferenze internazionali sul tema della sommarizzazione dei testi. Queste collezioni sono comunemente usate in letteratura per la valutazione dei metodi e offrono quindi una base comune per il confronto.

4.1.1 DUC2004

DUC (Document Understanding Conferences) [44] sono state una serie di conferenze sul tema dell'elaborazione automatica dei documenti tenute dal 2000 al 2007.

La pianificazione delle conferenze è stata ideata durante un meeting del progetto TIDES (Translingual Information Detection, Extraction and Summarization) del DARPA (Defense Advanced Research Projects Agency), un'agenzia del dipartimento americano della difesa. Lo scopo del progetto TIDES era quello di sviluppare le tecnologie necessarie per consentire ad operatori e analisti di madrelingua inglese di utilizzare una grossa mole di documenti che, al momento della creazione del progetto, non venivano analizzati per mancanza di analisti esperti e con conoscenze delle lingue straniere. Il progetto era votato allo sviluppo di tecnologie su quattro fronti principali: Traduzione (di materiale in altre lingue in inglese), Individuazione (trovare o scoprire informazioni necessarie come ad esempio argomenti), Estrazione (estrarre informazioni strutturate come entità e relazioni) e Sommarizzazione (ridurre la quantità di testo che un utente deve leggere) [11].

Nel corso delle conferenze venivano valutati gli strumenti sviluppati da vari ricercatori su compiti specifici pre-assegnati.

Come riferimento per il presente lavoro è stato preso il *task* 2 della conferenza DUC 2004, che consiste nella creazione di sommari brevi di cluster di documenti [46]. Per lo svolgimento sono forniti cinquanta cluster, di dieci documenti ciascuno, da cui generare

cinquanta riassunti da 665 byte. Ognuno dei cluster è formato da dieci articoli di giornale inerenti lo stesso argomento. Tutti i documenti sono in lingua inglese e con una lunghezza di riga inferiore agli ottanta caratteri. Questa caratteristica deve essere tenuta in considerazione nella fase di *sentence detection*.

> *Cambodia's bickering political parties broke a three-month deadlock Friday and agreed to a coalition government leaving strongman Hun Sen as sole prime minister, King Norodom Sihanouk announced. In a long-elusive compromise, opposition leader Prince Norodom Ranariddh will become president of the National Assembly resulting from disputed elections in July, even though Hun Sen's party holds a majority of 64 seats in the 122-member chamber. Hun Sen's Cambodian People's Party dropped insistence on a joint assembly chairmanship shared by Ranariddh and party boss Chea Sim, the current speaker. It was one of the main stumbling blocks in months of discord. Instead, Sihanouk announced, the constitution will be modified to create a new Senate, which Chea Sim will head. Chea Sim will still serve as acting head of state during the king's frequent absences from the country. "The major political crisis in the country has been resolved and the political deadlock facing the nation has also come to an end," the king said in his statement. The Senate will initially be appointed by the king.*
> …

Figura 4.1: DUC2004 - Estratto di un documento appartenente a un cluster

Per ognuno dei cluster vengono forniti uno o più riassunti manuali (*golden standard summary*) creati da diversi operatori.

Gli esempi riportati in figura 4.2 e 4.3 mostrano come i riassunti *human-generated* siano spesso difformi tra loro. Questo è tanto più vero per la sommarizzazione generica dove la mancanza di un funzione prestabilita per il riassunto lascia lo spazio alle interpretazioni personali dovute a caratteristiche culturali e personali del redattore.

Per la valutazione dei risultati nel corso del DUC2004 è stato utilizzato il pacchetto ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) [29], descritto più in dettaglio nel paragrafo 4.2.3.
Per la valutazione dei risultati ottenuti da LSA-itemset summarizer si è scelto di adottare lo stesso metodo.

Prospects were dim for resolution of the political crisis in Cambodia in October 1998.
Prime Minister Hun Sen insisted that talks take place in Cambodia while opposition leaders Ranariddh and Sam Rainsy, fearing arrest at home, wanted them abroad.
King Sihanouk declined to chair talks in either place.
A U.S. House resolution criticized Hun Sen's regime while the opposition tried to cut off his access to loans.
But in November the King announced a coalition government with Hun Sen heading the executive and Ranariddh leading the parliament.
Left out, Sam Rainsy sought the King's assurance of Hun Sen's promise of safety and freedom for all politicians.

Figura 4.2: DUC2004 - Riassunto manuale del cluster - operatore A

Cambodian prime minister Hun Sen rejects demands of 2 opposition parties for talks in Beijing after failing to win a 2/3 majority in recent elections.
Sihanouk refuses to host talks in Beijing.
Opposition parties ask the Asian Development Bank to stop loans to Hun Sen's government.
CCP defends Hun Sen to the US Senate.
FUNCINPEC refuses to share the presidency.
Hun Sen and Ranariddh eventually form a coalition at summit convened by Sihanouk.
Hun Sen remains prime minister, Ranariddh is president of the national assembly, and a new senate will be formed.
Opposition leader Rainsy left out.
He seeks strong assurance of safety should he return to Cambodia.

Figura 4.3: DUC2004 - Riassunto manuale del cluster - operatore B

4.1.2 TAC2011

Dal 2008 gli obiettivi del DUC sono stati inseriti all'interno del TAC (*Text Analysis Conference*), una serie di workshop di valutazione con lo scopo di alimentare la ricerca nell'ambito del *Natural Language Processing* e delle sue applicazioni. I workshop del TAC sono organizzati dal NIST (National Institute of Standards and Technology) parte del U.S. Department of Commerce. Ogni TAC viene suddivisa in vari argomenti (*track*) a loro volta suddivisi per obiettivi specifici (*task*).

I dati utilizzati nel presente lavoro sono riferiti al TAC 2011, *Summarization Track* [45] e, in particolare, al task *MultiLing Pilot* [18]. L'obiettivo del task era quello di valutare

metodi di sommarizzazione parzialmente o completamente indipendenti dalla lingua.

L'insieme dei documenti è stato creato in due fasi. Nella prima fase è stato raccolto un corpus in lingua inglese composto da dieci collezioni, basate su un argomento specifico e composte ciascuna da dieci documenti. Ogni documento è stato selezionato in modo che contenesse almeno una sequenza di eventi. Nella seconda fase il corpus è stato tradotto, secondo un approccio frase per frase, in altre sei lingue: arabo, ceco, francese, greco, ebraico, hindi. La struttura dei file è analoga a quella del DUC2004 ma replicata per ognuna delle lingue proposte. Nelle figure 4.4 e 4.5 sono mostrati gli estratti della stessa parte di un documento, appartenenti alle corrispondenti collezioni in inglese e in hindi. In generale nelle traduzioni, anche per fattori culturali, non sempre viene mantenuta la struttura delle frasi ma gli organizzatori del TAC2011 hanno preferito l'approccio frase per frase per mantenere una uniformità nella struttura tra le varie lingue, visto anche l'utilizzo preponderante di tecniche di tipo estrattivo.

7.0 magnitude earthquake strikes off Haitian coast
Tuesday, January 12, 2010
A 7.0 magnitude earthquake has struck off the coast of Haiti earlier today at 21:53 UTC, according to the US Geological Survey (USGS). According to the US National Oceanic and Atmospheric Administration, no tsunami warning was issued, contradicting some media reports that said there was one in place. The quake's magnitude was revised down from an initial report of 7.3 on the Richter scale.
"A destructive widespread tsunami threat does not exist based on historical earthquake and tsunami data," the US Pacific Tsunami Warning Center said. "However, there is the possibility of a local tsunami that could affect coasts located usually no more than a 100 km [60 miles] from the earthquake epicentre."
...

Figura 4.4: TAC2011 - Estratto di un documento in inglese

I sommari valutati dovevano essere di lunghezza compresa tra 240 e 250 parole ed erano valutati con metodi sia manuali sia automatici [10]. Come nel DUC2004, per ogni *cluster* erano forniti dei sommari manuali.
Gli esempi in figura 4.6 e 4.7 mostrano, anche in questa occasione, una differenziazione tra i riassunti manuali, in questo caso incentrati uno più sugli aspetti economici e uno più su quelli umanitari.

हैती ने ७.० मैगनिट्यूड का भुकंप आया।
मंगलवार, जनवरी १२, २०१०
अमरीका के भूवैग्यानिक सर्वेक्षण के अनुसार आज सुबह ७.०मैगनिट्यूड का भूकंप हैती में २१:५३ यू टी सी में आया। अमरीका के राष्ट्रीय समुद्रीय और वायुमंडलीय प्रशासन ने कहा कि सुनामी चेतावनी नहीं दी गई। कुछ मीडिया की खबरों में सुनामी के आने की खबर थी। भूकंप को शुरू में रिक्टर स्केल में ७.३ मैगनिट्यूड नापा गया जिसे बाद में कम कर दिया गया।
अमरीका के प्रशांत सुनामी चेतावनी केंद्र ने कहा कि बहुत बड़े सुनामी के आने कि आशंका नहीं है। ऐतिहासिक आंक्ड़े ऐसा न्हीं बताते। पर हो सक्ता है कि एक स्थानीय सुनामी आया,जो तटीय क्षेत्र से लगभग १०० कि.मी (६० मील) भूकंप उत्केंद्र से दूर है।
…

Figura 4.5: TAC2011 - Estratto del documento in figura 4.4, tradotto in hindi

The strong 7.0 magnitude earthquake that hit Haiti on 12 January left much of the nation in ruins. The epicenter was fifteen kilometers southwest of the capital, Port-au-Prince, at a depth of ten kilometers. At least 27 aftershocks were also recorded. Buildings across the capital have collapsed, including hospitals, the Parliament and the headquarters of the UN peacekeeping mission in Haiti. Communications and power were out across the city. A strong earthquake (6.1) struck once again Haiti on 20 January, as an international aid effort was underway to help those affected by the previous one. Eleven days after the initial earthquake, the rescue efforts were abandoned and focus was turned to the logistically difficult relief effort. Survivors have been living in makeshift camps. The Haitian president asked for 200,000 tents and 26 million ready-to-eat meals to be airdropped. WFP has provided approximately 2 million meals. Even as aid is flowing into the country, survivors face increasing insecurity from convicts escaped from collapsed prisons and human traffickers. Haitian police arrested ten US missionaries who tried to take 33 Haitian children out the country without permissions. Meanwhile, doctors are struggling to treat thousands of injured with limited resources. Lots of injured were flown to the US for medical care. Overall, the earthquake killed exactly 222.570 people and affected 3 million. The economic loss suffered by Haiti reached 7.754 billion dollars. The full reconstruction of the country, described as a "colossal work of reconstruction" by the Haitian Prime Minister, could take several decades.

Figura 4.6: TAC2011 - Riassunto manuale, in 250 parole, della collezione a cui appartiene il documento in figura 4.4

On January 12, 2010 a 7.0 magnitude earthquake struck off the coast of Haiti. Due to the earthquake's severity, information was initially confusing, but later the disaster's size was obvious. The first assessments were talking about thousands of deaths and millions of homeless people. According to government sources of Haiti, 60 percent of Haiti's gross domestic product was lost under the ruins and also, a large number of government officials and police and military forces were among the earthquake casualties. Thousands of dead remained unburied in the streets, and violence and crime were dramatically rose. On January 20, a 6.1 magnitude earthquake had once again struck Haiti, worsening the situation. The US, the UN and other countries said they would provide all necessary assistance to Haiti. After a period of ten days, it was announced that rescue efforts were going to abandon, although international rescue teams pulled out two people alive in the last days. At the end of the month, the US military had halted the evacuation of victims of the Haiti earthquake for medical care, reportedly due to uncertainty about who would pay for the costs, but immediately the US government announced that the evacuations flights would resume. After the earthquake, a major problem in Haiti was a number of alleged abduction of Haitian children for illegal adoption. At the border with the Dominican Republic 10 US missionaries were arrested, accused of child kidnapping. They tried to take 33 Haitian children out the country without the government's consent.

Figura 4.7: TAC2011 - Altro riassunto manuale del cluster contenente il documento in figura 4.4

Le valutazioni automatiche erano demandate al sistema AutoSumENG[1] [17] nella variante MeMoG [16], a ROUGE (in particolare alle varianti ROUGE1, ROUGE2, ROUGE-SU4) e a un sistema di valutazione manuale in cui degli operatori erano invitati ad assegnare un punteggio, da uno a cinque, sulla base della corrispondenza del riassunto con la collezione di documenti originali.

Giannakopoulos, il coordinatore del task, pone l'attenzione su una serie di problemi emersi nel tradurre i vari documenti (come ad esempio per la traduzione di nomi, acronimi e forme idiomatiche) e sull'opportunità, per future ricerche, di collezionare materiale originale delle lingua in analisi e non tradotto da una lingua di partenza [18].

[1] AutoSumENG è un sistema di valutazione automatica della qualità di un riassunto per comparazione con sommari manuali. Utilizza grafi di n-grammi di parole o caratteri ma supporta altri approcci basati su istogrammi.

An earthquake, with a magnitude of 7.0, struck Haiti on January 12, killing as much as 200,000 people and largely destroying the capital Port-au-Prince; another million have been left without homes.
The survivors from the recent 7.0 magnitude earthquake in Haiti are now facing increasing insecurity from human traffickers and convicts escaped from collapsed prisons, officials have cautioned, even as aid is flowing into the country.
A massive earthquake, registering 7.0 on the moment magnitude scale, struck Haiti yesterday, destroying many buildings, disrupting communications, and burying an unknown number of people underneath rubble.
A 7.0 magnitude earthquake has struck off the coast of Haiti earlier today at 21:53 UTC, according to the US Geological Survey (USGS).
A strong earthquake has once again struck Haiti, eight days after a 7.0 magnitude quake left much of the nation in ruins.
The United Nations has announced that the government of Haiti has put an end to its efforts to find and rescue buried survivors of the earthquake that hit the region eleven days ago.
Ten United States missionaries who tried to take 33 Haitian children out the country last week without the government's consent have been charged with child kidnapping and criminal association for illegally trying to take children out of Haiti.
6.1 magnitude aftershock earthquake hits Haiti
This quake is said to have been the strongest in Haiti in over two hundred years; the last time an earthquake of comparable magnitude was recorded was in 1770.

Figura 4.8: TAC2011 - Riassunto automatico tramite LSA-itemset summarizer del cluster contenente il documento in figura 4.4

4.2 Validazione

La valutazione di un metodo di sommarizzazione è un compito che presenta varie complessità. La qualità di un riassunto è determinata da fattori diversi:

- **Efficacia:** se utilizzato per uno scopo specifico, la qualità di un riassunto è data dalla sua efficacia per l'utilizzatore finale nel sostituire i documenti originali per lo svolgimento di un'operazione, con una riduzione della precisione che sia marginale o comunque giustificata dal guadagno in termini di tempo. Questo tipo di misurazione può essere effettuata su dei task specifici come ad esempio nel SUMMAC [36], descritto nel paragrafo 2.2. Per un sommarizzatore generico, il cui scopo non è determinato a priori, questo tipo di valutazione andrebbe fatta su un insieme

di possibili ambiti di applicazione, rendendo questo tipo di approccio in molti casi impercorribile.

- **Qualità linguistica:** in base all'utilizzo del riassunto potrebbe essere importante la sua qualità linguistica. In particolare per un riassunto di tipo estrattivo possono essere determinanti i metodi per la suddivisione del testo in frasi e quelli di selezione delle frasi stesse. Un riassunto potrebbe contenere tutte le informazioni necessarie ma contenere frasi prive di significato, ridondanti o ordinate male a livello semantico rendendolo di difficile lettura.

- **Qualità del contenuto:** Un riassunto dovrebbe contenere tutti e soli gli aspetti più rilevanti dei documenti di partenza.

Negli esperimenti effettuati è stata valutata principalmente la qualità del contenuto dei riassunti prodotti che è più direttamente e oggettivamente misurabile con i sistemi illustrati di seguito. Alcune considerazioni sono state fatte anche sulla qualità linguistica nella scelta di alcuni dettagli implementativi.

4.2.1 Valutazione manuale e automatica

Nel corso delle conferenze del NIST sono stati utilizzati vari metodi, sia automatici sia manuali, per la valutazione dei riassunti che sono diventati standard di fatto per la presentazione dei risultati di nuove pubblicazioni [32].

La valutazione manuale viene effettuata da valutatori esperti che utilizzano due metodi principali. Uno consiste nel dare un voto diretto in una scala (es. 1-5 o 1-10) alla qualità del sommario, l'altro nel compararlo con un riassunto detto *gold-standard*, un modello creato a sua volta da un esperto e che possa considerarsi un riferimento. L'utilizzo di modelli di riferimento creati da non esperti inficia la qualità della valutazione finale, introducendo una maggiore quantità di elementi soggettivi. Allo stesso modo l'utilizzo di valutatori non esperti per la comparazione di sommari con i *gold standard* influenza la valutazione. Gillick e Liu hanno dimostrato come l'identità del valutatore diventasse l'elemento più determinante nel punteggio finale nel caso di valutatori non esperti [19].

Un metodo basato sull'utilizzo dei modelli, e uno dei primi ad essere stato utilizzato, è quello di dare un punteggio in base al grado di copertura del contenuto. I risultati di questo metodo erano però troppo fortemente influenzati dal modello utilizzato come riferimento.

Un metodo successivo, detto metodo di valutazione piramidale, cerca di risolvere questo problema. Nel metodo piramidale vengono utilizzati dei modelli *human generated* in cui vengono annotate delle parti significative semanticamente, dette *Summary Content Units* (SCU). Ad ogni SCU viene assegnato un peso pari al numero di modelli che la riportano. Un riassunto ideale dovrebbe contenere un sottoinsieme di tutti gli SCU, formato

da quelli con indice più elevato. Il *pyramid score* per un sommario S è identificato dal seguente rapporto:

$$\text{py(S)} = \frac{\text{somma dei pesi degli SCU presenti in S}}{\begin{array}{c}\text{somma dei pesi di un sommario ideale} \\ \text{contenente lo stesso numero di SCU di S}\end{array}} \tag{4.1}$$

In questo modo si ottiene uno *score* più affidabile, basato su molteplici sommari di riferimento. Nei TAC sono stati solitamente utilizzati quattro *gold-standard* per la valutazione piramidale [32].

Per la valutazione della qualità generale del riassunto, comprensiva sia della qualità del contenuto affrontata già dai metodi precedenti sia della qualità linguistica, viene utilizzato anche un altro indice detto *responsiveness*.

La valutazione manuale richiede un impiego di risorse non indifferente. Come detto in precedenza è importante che le fasi di preparazione dei *gold standard* e di valutazione siano effettuate da operatori esperti e preparati ma, nel caso di valutazione manuale, devono essere effettuate da esperti anche le fasi di annotazione dei documenti e di valutazione. E' stato stimato che ad esempio l'esecuzione di queste fasi per un DUC poteva richiedere più di 3000 ore uomo [29]. Un sistema di valutazione automatica permette un risparmio di risorse e al contempo la possibilità di replicare e confrontare i risultati, fattore determinante nella ricerca.

4.2.2 *Precison*, *Recall* e *F-score*

Per cercare di effettuare una valutazione automatica della qualità di un riassunto, confrontandolo con dei modelli di riferimento, è necessario individuare degli indicatori. Per questo scopo i sistemi di valutazione utilizzano spesso i concetti di *Precison* e *Recall*.

Supponendo di applicare queste metriche ai termini, la *recall* indica quanti dei termini presenti nel modello sono presenti anche nel riassunto in esame:

$$R = \frac{\text{Numero di parole presenti sia nel riassunto sia nel modello}}{\text{Numero di parole presenti nel modello}} \tag{4.2}$$

Questa metrica da sola non tiene però conto di quanti termini ha introdotto il riassunto e che non sono presenti nel modello. In questo senso ad esempio il testo originale avrà un valore elevato di *recall* ma non può certo considerarsi un buon riassunto. La *precision* ha l'obiettivo di individuare questo secondo aspetto ed è definita come segue:

$$P = \frac{\text{Numero di parole presenti sia nel riassunto sia nel modello}}{\text{Numero di parole presenti nel riassunto}} \tag{4.3}$$

In questo modo l'indicatore per un riassunto 'prolisso' risulterà di valore modesto.

Per cercare di mediare tra questi due fattori si introduce un indice, chiamato *F-measure* o *F-score*, che rappresenta la media armonica tra i due indici precedenti:

$$F = F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{4.4}$$

La formula 4.4 dà la stessa rilevanza ai due fattori e può essere vista come un caso particolare, con $\beta = 1$, della formula più generica:

$$F\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R}, \quad \beta \in \mathbb{R}^+ \tag{4.5}$$

nella quale col valore di β è possibile modulare la rilevanza rispettiva di *precision* e *recall*.

4.2.3 Rouge

Introdotto come uno dei sistemi di valutazione del DUC 2004, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [29] è uno degli strumenti di riferimento ancora oggi per la valutazione della qualità dei riassunti.
ROUGE è uno strumento per il calcolo di una serie di metriche di valutazione, basate sulla presenza di elementi comuni nel riassunto e nei modelli, quali *n-gram*[2], sequenze o coppie di termini. Le metriche calcolate sono:

- **Rouge-N (N-gram Co-Occurrence Statistics):** Rouge-N è una misura di *recall* tra il sommario in esame e un set di modelli. Formalmente è definta come:

$$\text{Rouge-N} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{4.6}$$

  dove n è la lunghezza dell'*n-gram*, $gram_n$ e $Count_{match}(gram_n)$ sono il massimo numero di n-gram che co-occorrono nel sommario in esame e in un set di modelli di riferimento. Il numero di *n-gram* al denominatore aumenta se vengono presi in considerazione più modelli di riferimento. Questo comportamento è desiderabile visto che sono possibili diversi riassunti qualitativamente buoni. Al numeratore vengono sommate tutte le corrispondenze ai vari modelli, per cui vengono favoriti riassunti che riportano *n-gram* presenti in più modelli [29].

[2]Un *n-gram* è un insieme di n termini che compaiono consecutivamente nella stessa unità di testo (es. frase o paragrafo)

- **Rouge-L (Longest Common Subsequence):** Per il calcolo di questo indice le frasi possono essere viste come sequenze di termini. La LCS è la sequenza più lunga in comune tra due frasi. Il metodo calcola un *F-score* basato su una combinazione di LCS a livello di frase e di riassunto.

- **Rouge-W (Weighted Longest Common Subsequence):** Questo indice è simile al precedente ma tiene in conto anche le distanze relative tra termini all'interno di una sequenza.

- **Rouge-S (Skip-Bigram Co-Occurrence Statistics):** Uno *skip-bigram* è una coppia di parole presenti all'interno di una frase riportate nel loro ordine originale, indipendentemente dai termini presenti tra di esse. Dato un *gold-standard* X di lunghezza m e una traduzione da valutare Y di lunghezza n, l'indice di *F-score* basato *skip-bigrams* può essere calcolato come segue:

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \tag{4.7}$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \tag{4.8}$$

$$F_{skip2} = \frac{(1+\beta^2) \cdot R_{skip2} \cdot P_{skip2}}{R_{skip2} + \beta^2 \cdot P_{skip2}} \tag{4.9}$$

  dove $SKIP2(X,Y)$ è il numero di *skip-bigram* in comune tra X e Y, β un fattore di controllo dell'importanza relativa di P_{skip2} e R_{skip2} e C una funzione di combinazione definita come $C(a,b) = a!/(b! * b!)$. Il vantaggio di questo indice è che non richiede corrispondenze consecutive pur tenendo conto dell'ordine in cui i termini appaiono. Per evitare che vengano considerate positive delle corrispondenze casuali è possibile limitare il massimo numero di termini che può intercorrere tra i due elementi del *bigram*.

- **Rouge-SU** è un'estensione di Rouge-S che ha il problema di valutare zero una frase in cui non ci siano co-occorrenze di *skip-bigrams* con il riferimento anche in presenza di co-occorrenza di molti termini. In questo indice si tengono in conto anche le occorrenze di *unigram*. Il numero massimo di termini che può frapporsi tra i due elementi dei *bi-gram* sono convenzionalmente indicati nel nome del metodo. Rouge-SU4 ad esempio utilizza *unigram* e *bi-gram* con uno *skip* di massimo 4 termini.

L'autore ha eseguito vari test sui dati dei DUC degli anni precedenti il 2004 sugli indici di ROUGE confrontandoli con i metodi di valutazione manuale. La correlazione con *Pyramid* ad esempio è risultata molto elevata (0,94 con ROUGE-2 e 0,92 con ROUGE-SU4) confermandone la validità.

4.2.4 Altri metodi di valutazione automatica

Metodi che prevedono l'utilizzo di *gold standard*

ROUGE presenta alcuni limiti legati alla sua implementazione. La mancanza del supporto per i caratteri Unicode e il *tokenizer* basato su espressioni regolari possono creare risultati meno attendibili per le lingue diverse dall'inglese. JRouge [3] e ROUGE 2.0 [4] sono dei *porting* di Rouge in linguaggio Java. Queste implementazioni consentono la corretta interpretazione dei caratteri Unicode ma per entrambe sono disponibili solamente le metriche ROUGE-N.
Un altro sistema di valutazione noto è AutoSummENG (AUTOmatic SUMMary Evaluation based on N-gram Graphs) [17]. Il metodo è basato su grafi di *n-gram* ed stato costruito cercando di soddisfare tre caratteristiche:
– indipendenza rispetto alla lingua;
– completa automazione, il metodo non richiede interventi umani ulteriori rispetto all'uso di sommari di riferimento;
– sensibilità verso il contesto.
AutosummENG si è rivelato molto affidabile anche per le lingue diverse dall'inglese.

Metodi che non prevedono l'utilizzo di *gold standard*

I metodi visti finora prevedono l'utilizzo di riassunti di riferimento. Sebbene questo approccio abbia dimostrato la sua validità, richiede l'onerosa produzione dei *gold standard* o, in alternativa, di limitare l'ambito di test alle collezioni per cui i riassunti umani sono già stati prodotti. Steinberger e Ježek, nello stesso lavoro in cui propongono un sommarizzatore basato su LSA [52], presentano anche un metodo di valutazione basato sulla stessa tecnica. Il metodo prevede di eseguire la SVD sia sul testo originale sia sul riassunto per poi confrontare la rilevanza dei termini dei due testi, calcolata come un *salience score* dei termini a partire dalle matrici U e Σ.

Annie Louis e Ani Nenkova hanno sviluppato vari approcci per la valutazione della bontà di un sommario, basati sulla similarità nella distribuzione dei termini tra originale e riassunto. La migliore delle metriche individuate, la divergenza di *Jensen-Shannon*, ha una correlazione di 0.9 con le valutazioni manuali nei test condotti [31]. Un'altra tecnica introdotta dalle autrici è la generazione di pseudomodelli che vanno ad aggiungersi a eventuali modelli umani per migliorare le prestazioni dei sistemi di valutazione come

[3] JRouge: `https://bitbucket.org/nocgod/jrouge/wiki/Home`

[4] Rouge 2.0: `http://www.rxnlp.com/rouge-2-0/`

ROUGE. [32]. L'implementazione dei vari metodi è stata resa disponibile nel software SIMetrics (Summary Input similarity Metrics)[5].

4.3 Risultati

Nelle tabelle riassuntive sono state riportate le comparazioni con i metodi i cui risultati sono reperibili per le stesse collezioni e il metodo LSA-itemset summarizer nelle seguenti tre varianti:

- **LSA-i-FI**: LSA-itemset summarizer con *frequent itemset*
- **LSA-i-S**: LSA-itemset summarizer con *n-gram*
- **LSA-i-FIS**: LSA-itemset summarizer con *frequent itemset* di *n-gram*

Per ogni collezione e per ogni lingua il metodo è stato provato con diverse combinazioni dei parametri descritti nel paragrafo 3.2. Alcuni parametri sono stati impostati allo stesso valore per tutti i test, come ad esempio la dimensione massima della matrice, fissata a 1GB. Il parametro *ItemsetType* è relativo al metodo di estrazione dei *frequent itemset* e non è applicabile alla versione del metodo che prevede l'estrazione di *n-gram*.
I risultati sono relativi alla combinazione sperimentalmente più efficace.

I valori riportati nelle tabelle sono le medie dei risultati ottenuti dai metodi sulle collezioni del dataset. Le colonne contengono i valori medi di *recall*, *precision* e *F1-score* relativi ai metodi ROUGE-2 e ROUGE-SU4 e sono ordinati per il valore di ROUGE-SU4 F1-score decrescente. I valori migliori di ogni indice sono stati evidenziati in neretto.

4.3.1 Risultati sul dataset DUC2004

Le prestazioni sulle collezioni del DUC2004 sono state confrontate con i risultati dei metodi che hanno partecipato alla competizione e di altri metodi sviluppati dopo il 2004. I risultati del metodo nella versione con *frequent itemset* sulle collezioni del DUC2004, sono paragonabili ai migliori metodi precedenti su tutti gli indici mentre le altre due varianti ottengono risultati inferiori.

[5] SIMetrics: `http://homepages.inf.ed.ac.uk/alouis/IEval2.html`

Tabella 4.1: Parametri utilizzati per la collezione DUC2004

	ItemsetType	**minSup**	**minItemsetLength**	**similarityThreshold**
LSA-i-FI	closed	0.03	1	0.8
LSA-i-S	NA	0.03	1	0.7
LSA-i-FIS	all	0.03	1	0.7

Tabella 4.2: Risultati del pacchetto ROUGE sulla collezione DUC2004

	ROUGE-2			**ROUGE-SU4**		
Summarizer	**R**	**Pr**	**F1**	**R**	**Pr**	**F1**
LSA-i-FI	0.0916	0.0923	0.0919	**0.1343**	0.1353	**0.1347**
DUC'04CLASSY-Serifpeer67	0.0906	**0.0941**	**0.0922**	0.1313	**0.1362**	0.1335
DUC'04CLASSY-prepeer65	**0.0922**	0.0909	0.0915	0.1335	0.1313	0.1323
MWI-Sum	0.0904	0.0916	0.0909	0.1312	0.1328	0.1319
DUC'04CLASSY-baselinepeer66	0.0888	0.0936	0.0909	0.1284	0.1352	0.1313
ICSIsumm	0.0877	0.0861	0.0869	0.1310	0.1285	0.1297
DUC'04peer35	0.0837	0.0842	0.0839	0.1288	0.1297	0.1292
DUC'04peer104	0.0857	0.0842	0.0849	0.1294	0.1270	0.1281
DUC'04peer102	0.0848	0.0859	0.0853	0.1273	0.1286	0.1278
DUC'04peer124	0.0833	0.0819	0.0826	0.1278	0.1253	0.1265
ItemSum	0.0852	0.0870	0.0859	0.1254	0.1276	0.1262
LSA-i-S	0.0863	0.0871	0.0866	0.1255	0.1266	0.1260
DUC'04peer19	0.0803	0.0804	0.0803	0.1247	0.1247	0.1246
DUC'04peer81	0.0808	0.0790	0.0799	0.1253	0.1224	0.1238
DUC'04peer34	0.0763	0.0764	0.0763	0.1236	0.1240	0.1238
LSA-i-FIS	0.0834	0.0841	0.0837	0.1227	0.1239	0.1232
OTS	0.0744	0.0740	0.0742	0.1151	0.1144	0.1147
TexLexAn	0.0658	0.0655	0.0656	0.1096	0.1088	0.1092
AMTS	0.0635	0.0651	0.0642	0.1014	0.1040	0.1025

4.3.2 Risultati sul dataset TAC2011

Anche per il TAC2011 il metodo LSA-itemset summarizer è stato messo a confronto con i metodi partecipanti alla conferenza, in particolare al task *MultiLing Pilot*, per il quale non tutti i *competitor* hanno partecipato per tutte le lingue, e con altri metodi più recenti.

Arabo

Per la lingua araba tutte le versioni di LSA-itemset summarizer e in particolare quelle basate su *n-gram*, ottengono risultati migliori rispetto agli altri metodi, in alcuni casi con margine molto elevato, soprattutto per quanto riguarda la *precision*. Gli ottimi risultati nell'utilizzo degli *n-gram* non si ripetono in tutte le lingue ma indicano che, con un'opportuna configurazione, questo approccio può essere efficace.

Tabella 4.3: Parametri utilizzati per la collezione TAC2011, arabo

	ItemsetType	**minSup**	**minItemsetLength**	**similarityThreshold**
LSA-i-FI	all	0.02	2	0.8
LSA-i-S	NA	0.03	3	0.9
LSA-i-FIS	closed	0.03	2	0.8

Tabella 4.4: Risultati del pacchetto ROUGE sulla collezione TAC2011, arabo

Summarizer	**R**	**Pr**	**F1**	**R**	**Pr**	**F1**
LSA-i-S	0.0886	0.2250	0.1210	**0.1155**	**0.4423**	**0.1654**
LSA-i-FIS	**0.1073**	**0.2291**	**0.1434**	0.1063	0.2734	0.1438
LSA-i-FI	0.1029	0.1871	0.1246	0.1080	0.2243	0.1270
UoEssex	0.0834	0.1289	0.0982	0.0993	0.1983	0.1263
AMTS	0.0812	0.1316	0.0969	0.0968	0.2020	0.1247
ItemSum	0.0851	0.1227	0.0952	0.0883	0.1526	0.1022
LIF	0.0717	0.1135	0.0775	0.0769	0.1535	0.0863
MWI-Sum	0.0513	0.2210	0.0756	0.0569	0.2908	0.0839
JRC	0.0454	0.1338	0.0519	0.0633	0.1965	0.0714
CLASSY	0.0605	0.1221	0.0719	0.0543	0.2528	0.0711
ICSIsumm	0.0487	0.1035	0.0626	0.0523	0.1216	0.0665
TALN_UPF	0.0337	0.0702	0.0445	0.0418	0.1362	0.0602
CIST	0.0502	0.0708	0.0578	0.0378	0.0825	0.0482
UBSummarizer	0.0092	0.0671	0.0161	0.0107	0.1375	0.0195

Ceco

Per il ceco il risultato migliore è ottenuto dalla versione di LSA-itemset summarizer che utilizza la combinazione di *n-gram* e *itemset* frequenti, con risultati migliori rispetto ai precedenti su tutti gli indici.

Tabella 4.5: Parametri utilizzati per la collezione TAC2011, ceco

	ItemsetType	**minSup**	**minItemsetLength**	**similarityThreshold**
LSA-i-FI	closed	0.02	3	0.7
LSA-i-S	NA	0.02	1	1
LSA-i-FIS	closed	0.02	1	0.8

Tabella 4.6: Risultati del pacchetto ROUGE sulla collezione TAC2011, ceco

Summarizer	**R**	**Pr**	**F1**	**R**	**Pr**	**F1**
LSA-i-FIS	**0.0951**	**0.2739**	**0.1410**	**0.1006**	**0.2910**	**0.1493**
CIST	0.0924	0.2567	0.1359	0.1001	0.2807	0.1475
MWI-Sum	0.0936	0.2576	0.1372	0.0975	0.2710	0.1433
CLASSY	0.0887	0.2477	0.1306	0.0960	0.2697	0.1415
JRC	0.0867	0.2465	0.1282	0.0947	0.2716	0.1403
ItemSum	0.0840	0.2460	0.1252	0.0905	0.2653	0.1349
LSA-i-S	0.0797	0.2193	0.1168	0.0915	0.2554	0.1346
ICSIsumm	0.0763	0.2206	0.1132	0.0872	0.2534	0.1296
LSA-i-FI	0.0788	0.2185	0.1158	0.0867	0.2427	0.1277
OTS	0.0719	0.1997	0.1057	0.0826	0.2313	0.1217
LIF	0.0748	0.2173	0.1113	0.0816	0.2390	0.1216
AMTS	0.0697	0.3087	0.1125	0.0718	0.3156	0.1158
UBSummarizer	0.0444	0.1276	0.0658	0.0638	0.1840	0.0947

Inglese

Per la lingua inglese il metodo ha buoni risultati, sia nella versione con *itemset* frequenti sia in quella con la combinazione di *n-gram* e *itemset* frequenti, pur non essendo il migliore in nessuno degli indici.

Tabella 4.7: Parametri utilizzati per la collezione TAC2011, inglese

	ItemsetType	**minSup**	**minItemsetLength**	**similarityThreshold**
LSA-i-FI	closed	0.02	2	0.9
LSA-i-S	NA	0.04	1	1
LSA-i-FIS	all	0.04	3	0.7

Tabella 4.8: Risultati del pacchetto ROUGE sulla collezione TAC2011, inglese

	ROUGE-2			**ROUGE-SU4**		
Summarizer	**R**	**Pr**	**F1**	**R**	**Pr**	**F1**
MWI-Sum	**0.1086**	**0.2382**	**0.1490**	**0.1204**	**0.2669**	**0.1658**
LSA-i-FI	0.0922	0.2056	0.1273	0.1157	0.2604	0.1601
ICSIsumm	0.0966	0.2191	0.1340	0.1105	0.2526	0.1537
LSA-i-FIS	0.0856	0.1920	0.1183	0.1094	0.2459	0.1513
CLASSY	0.0895	0.2038	0.1244	0.1076	0.2468	0.1498
JRC	0.0959	0.2224	0.1339	0.1053	0.2467	0.1475
ItemSum	0.0927	0.2137	0.1293	0.1056	0.2446	0.1474
LSA-i-S	0.0811	0.1817	0.1121	0.0992	0.2258	0.1377
LIF	0.0801	0.1809	0.1109	0.0978	0.2239	0.1361
TALN_UPF	0.0770	0.1719	0.1063	0.0984	0.2210	0.1361
OTS	0.0778	0.1798	0.1085	0.0936	0.2172	0.1307
CIST	0.0740	0.1663	0.1024	0.0915	0.2080	0.1271
TexLexAn	0.0713	0.1666	0.0998	0.0891	0.2101	0.1251
AMTS	0.0700	0.1629	0.0979	0.0849	0.1997	0.1191
SIEL_IIITH	0.0682	0.1527	0.0941	0.0859	0.1941	0.1189
UoEssex	0.0703	0.1625	0.0981	0.0848	0.1981	0.1187
UBSummarizer	0.0445	0.0981	0.0612	0.0725	0.1604	0.0998

Francese

Per il francese il metodo con gli *itemset* frequenti ottiene risultati lievemente migliori dei predecessori su tutti gli indici. Le altre due varianti hanno invece risultati inferiori.

Tabella 4.9: Parametri utilizzati per la collezione TAC2011, francese

	ItemsetType	**minSup**	**minItemsetLength**	**similarityThreshold**
LSA-i-FI	closed	0.03	3	0.7
LSA-i-S	NA	0.02	1	0.7
LSA-i-FIS	all	0.04	3	0.8

Tabella 4.10: Risultati del pacchetto ROUGE sulla collezione TAC2011, francese

	ROUGE-2			**ROUGE-SU4**		
Summarizer	**R**	**Pr**	**F1**	**R**	**Pr**	**F1**
LSA-i-FI	**0.1074**	**0.2506**	**0.1502**	**0.1185**	**0.2794**	**0.1663**
MWI-Sum	0.1071	0.2479	0.1494	0.1183	0.2759	0.1654
ICSIsumm	0.1021	0.2400	0.1432	0.1146	0.2718	0.1611
JRC	0.1029	0.2399	0.1439	0.1114	0.2622	0.1562
CLASSY	0.1024	0.2337	0.1424	0.1112	0.2563	0.1550
CIST	0.0947	0.2175	0.1318	0.1100	0.2544	0.1534
AMTS	0.0961	0.2446	0.1375	0.1052	0.2692	0.1508
LSA-i-FIS	0.0951	0.2283	0.1342	0.1057	0.2567	0.1496
LIF	0.0905	0.2169	0.1276	0.1057	0.2556	0.1495
ItemSum	0.0891	0.2082	0.1248	0.1018	0.2396	0.1429
TexLexAn	0.0901	0.2026	0.1247	0.1008	0.2279	0.1397
LSA-i-S	0.0816	0.1999	0.1158	0.0947	0.2331	0.1345
OTS	0.0818	0.1914	0.1145	0.0931	0.2181	0.1304
UBSummarizer	0.0729	0.1693	0.1019	0.0926	0.2165	0.1297
TALN_UPF	0.0619	0.1451	0.0867	0.0859	0.2040	0.1209
SIEL IIITH	0.0621	0.1465	0.0871	0.0838	0.1978	0.1176

Greco

Il greco è la lingua per la quale i risultati del metodo sono peggiori sia in termini assoluti sia nel paragone con gli altri metodi.

Tabella 4.11: Parametri utilizzati per la collezione TAC2011, greco

	ItemsetType	**minSup**	**minItemsetLength**	**similarityThreshold**
LSA-i-FI	closed	0.02	2	0.8
LSA-i-S	NA	0.03	3	0.9
LSA-i-FIS	closed	0.04	3	0.7

Tabella 4.12: Risultati del pacchetto ROUGE sulla collezione TAC2011, greco

	ROUGE-2			**ROUGE-SU4**		
Summarizer	**R**	**Pr**	**F1**	**R**	**Pr**	**F1**
ICSIsumm	**0.0940**	0.1673	**0.1167**	**0.0904**	0.2004	**0.1161**
AMTS	0.0659	0.1316	0.0850	0.0787	0.1879	0.1047
CLASSY	0.0749	0.1961	0.1032	0.0662	0.2141	0.0937
LSA-i-FI	0.0679	0.2161	0.0959	0.0541	0.2308	0.0781
MWI-Sum	0.0473	**0.2233**	0.0751	0.0462	**0.3361**	0.0744
LIF	0.0466	0.0717	0.0548	0.0543	0.1040	0.0679
ItemSum	0.0531	0.1139	0.0701	0.0441	0.1327	0.0623
LSA-i-FIS	0.0368	0.0339	0.0347	0.0596	0.0637	0.0577
OTS	0.0353	0.1443	0.0545	0.0298	0.1658	0.0467
CIST	0.0229	0.0890	0.0348	0.0243	0.1322	0.0378
JRC	0.0240	0.0834	0.0363	0.0241	0.0924	0.0356
LSA-i-S	0.0379	0.0410	0.0379	0.0328	0.0372	0.0333
UBSummarizer	0.0065	0.0331	0.0109	0.0119	0.1369	0.0209

Hindi

Per l'hindi i risultati sono allineati coi migliori predecessori per gli indici di ROUGE-SU4, mentre ottengono risultati decisamente migliori su ROUGE-2.

Tabella 4.13: Parametri utilizzati per la collezione TAC2011, hindi

	ItemsetType	**minSup**	**minItemsetLength**	**similarityThreshold**
LSA-i-FI	all	0.02	3	0.8
LSA-i-S	NA	0.04	1	0.7
LSA-i-FIS	closed	0.02	2	0.9

Tabella 4.14: Risultati del pacchetto ROUGE sulla collezione TAC2011, hindi

	ROUGE-2			**ROUGE-SU4**		
Summarizer	**R**	**Pr**	**F1**	**R**	**Pr**	**F1**
LSA-i-FIS	**0.0584**	**0.1331**	**0.0733**	0.0589	**0.1443**	**0.0696**
MWI-Sum	0.0417	0.0477	0.0404	**0.0760**	0.1000	0.0688
ICSIsumm	0.0500	0.0501	0.0486	0.0673	0.0786	0.0681
LSA-i-FI	0.0584	0.0583	0.0572	0.0477	0.0437	0.0435
JRC	0.0249	0.0166	0.0199	0.0412	0.0651	0.0326
SIEL_IIITH	0.0084	0.0112	0.0096	0.0206	0.0335	0.0255
ItemSum	0.0498	0.0332	0.0398	0.0356	0.0192	0.0249
LSA-i-S	0.0249	0.0332	0.0285	0.0142	0.0221	0.0173
AMTS	0	0	0	0.0142	0.0052	0.0077
CIST	0	0	0	0	0	0
CLASSY	0	0	0	0	0	0
LIF	0	0	0	0	0	0
UBSummarizer	0	0	0	0	0	0
TALN_UPF	0	0	0	0	0	0

Le tabelle 4.15 e 4.16 riassumono i risultati di LSA-itemset summarizer in relazione ai metodi confrontati e in assoluto. Dal paragone tra le due tabelle si può notare come l'incostanza nei valori di classifica sia spesso dovuta alle diverse prestazioni dei *competitor* e come i risultati siano invece abbastanza omogenei per gruppi di lingue.

In generale le prestazioni sono molto meno buone per greco e hindi rispetto alle altre lingue, con valori di F1-Score di circa la metà, ma per l'hindi il metodo LSA-i-FIS risulta comunque il migliore. I risultati di inglese e francese sono molto simili per tutti i metodi. I due metodi che utilizzano gli *n-gram* hanno output analoghi per ceco, inglese e francese e in maniera minore anche per l'arabo. Il fatto che per l'arabo si abbiano, rispetto al greco, risultati più vicini alle altre lingue europee, è un indice di come il metodo sia probabilmente sensibile alle differenze linguistiche più che all'alfabeto utilizzato. Sarebbe utile analizzare quanto la fase di elaborazione iniziale influenzi questi risultati. Non si può inoltre escludere che alcune variabilità nelle prestazioni siano dovute al modello di stesura dei *golden summary*, che sono stati prodotti da più esperti madrelingua (da uno a quattro per ogni lingua). Questo fa sì che al fattore personale umano si aggiungano dei fattori socio-culturali che possono avere una forte influenza sulle scelte effettuate soprattutto visto che gli argomenti proposti sono notizie internazionali.

Tabella 4.15: Posizione di LSA-itemset summarizer rispetto ai metodi presi a paragone nei risultati sulle collezioni del TAC2011, ordinati per ROUGE-SU4 F1-Score

	Arabo	**Ceco**	**Inglese**	**Francese**	**Greco**	**Hindi**
LSA-i-FI	3°	9°	**2°**	**1°**	**4°**	4°
LSA-i-S	**1°**	7°	8°	13°	12°	8°
LSA-i-FIS	2°	**1°**	4°	9°	8°	**1°**

Tabella 4.16: Riepilogo dei risultati ottenuti da LSA-itemset summarizer sulle collezioni del TAC2011, ROUGE-SU4 F1-Score

	Arabo	**Ceco**	**Inglese**	**Francese**	**Greco**	**Hindi**
LSA-i-FI	0.1270	0.1277	**0.1601**	**0.1663**	**0.0781**	0.0435
LSA-i-S	**0.1654**	0.1346	0.1377	0.1345	0.0333	0.0173
LSA-i-FIS	0.1438	**0.1493**	0.1513	0.1496	0.0577	**0.0696**

4.4 Effetto dei parametri

Per ognuno dei parametri del metodo sono stati utilizzati da due a quattro valori diversi. Vista l'elevata numerosità delle combinazioni possibili, per cercare di analizzare l'andamento dei risultati, sono stati fissati alcuni parametri e valutato il comportamento dell'indicatore ROUGE-SU4 F1-Score. Nelle figure 4.9, 4.10, 4.11 e 4.12 sono mostrati i risultati ottenuti sulle collezioni del TAC2011, fissando il valore di soglia per la similarità all'80% e la versione *all* di LCM. Le figure mostrano l'andamento al variare del supporto minimo e della lunghezza minima dell'*itemset*. Eventuali valori mancanti sono dovuti al fatto che la combinazione non ha prodotto nessun *itemset*/*n-gram*.

Per la lingua inglese i risultati mostrano come l'influenza dei parametri sia diversa per i tre metodi. Per il metodo con gli *itemset* frequenti il parametro più rilevante sembra essere la lunghezza minima dei *itemset* anche se un supporto troppo elevato porta ad un degrado delle prestazioni, fino alla mancanza di output se combinato con una lunghezza minima elevata.

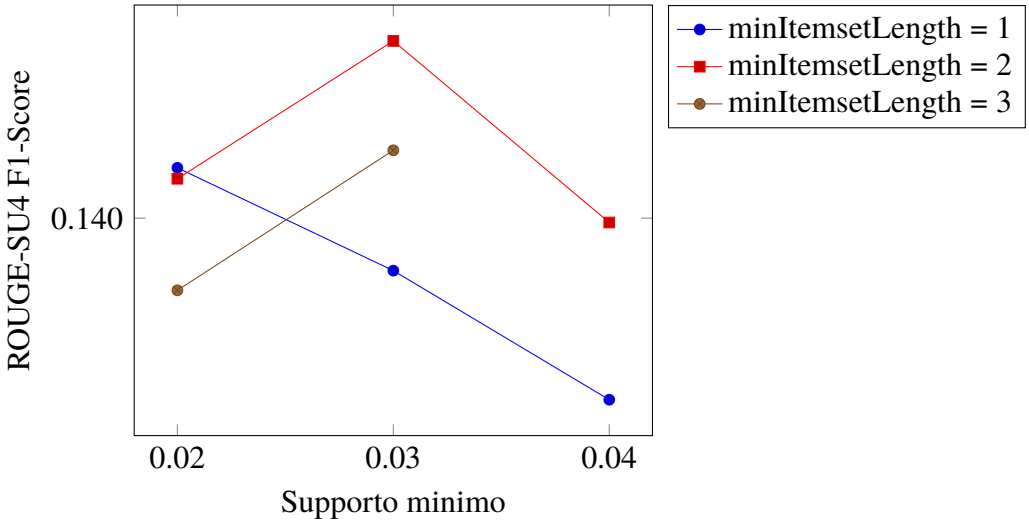


Figura 4.9: LSA-i-FI - Inglese - Confronto dell'andamento dell'indice ROUGE-SU4 F1-Score in funzione della variazione di supporto minimo e lunghezza dell'*itemset*.

Il metodo basato sugli *n-gram* per l'inglese ottiene risultati migliori includendo anche gli *unigram* e dà risultati analoghi per le due configurazioni con supporto più elevato.

Nel metodo combinato si nota invece una forte correlazione dei risultati con il supporto minimo, mentre la lunghezza degli itemset non sembra influenzare il risultato.

Il comportamento osservato in figura 4.11 tuttavia è solo in parte replicato nelle altre lingue. Ad esempio per il francese, che nei risultati ottenuti è in generale la lingua che più si avvicina all'inglese, conferma la scarsa dipendenza dalla lunghezza minima dell'*itemset* ma ha un andamento diverso rispetto al supporto minimo.

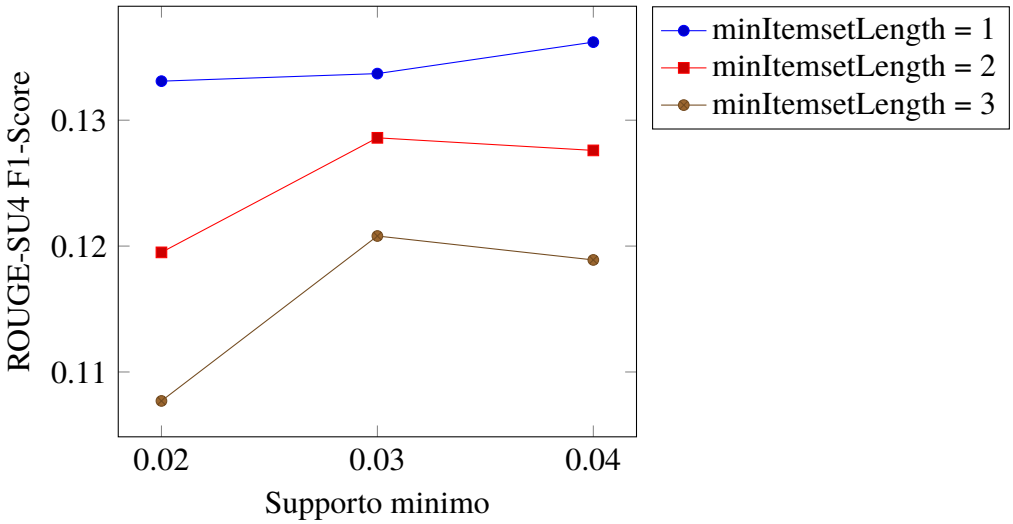


Figura 4.10: LSA-i-S - Inglese - Confronto dell'andamento dell'indice ROUGE-SU4 F1-Score in funzione della variazione di supporto minimo e lunghezza dell'*itemset*.

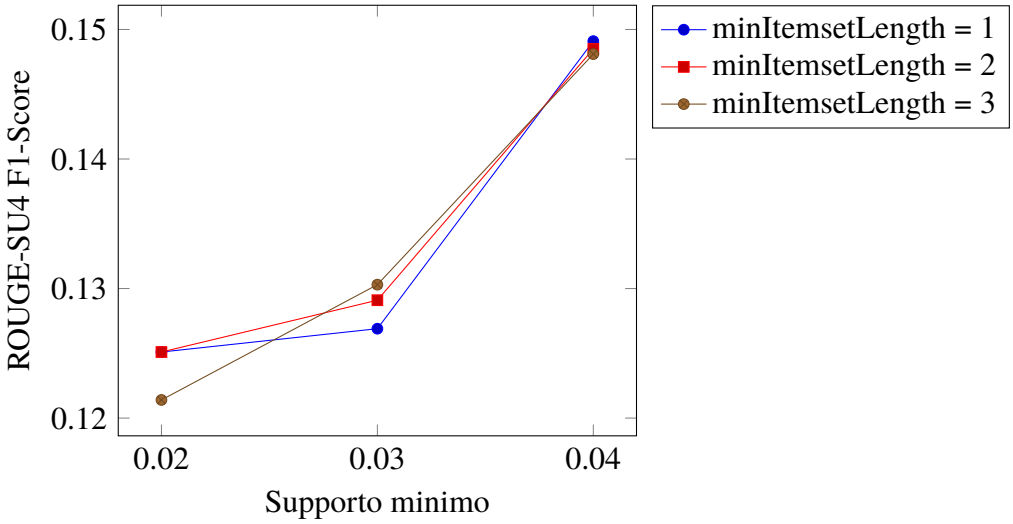


Figura 4.11: LSA-i-FIS - Inglese - Confronto dell'andamento dell'indice ROUGE-SU4 F1-Score in funzione della variazione di supporto minimo e lunghezza dell'*itemset*.

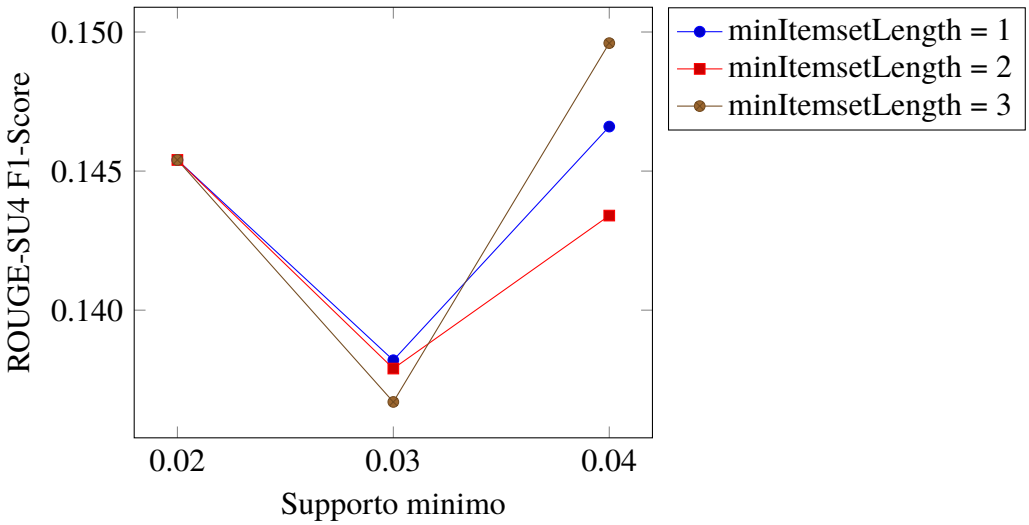


Figura 4.12: LSA-i-FIS - Francese - Confronto dell'andamento dell'indice ROUGE-SU4 F1-Score in funzione della variazione di supporto minimo e lunghezza dell'*itemset*.

Capitolo 5

Conclusioni

Gli indici ottenuti nei risultati sperimentali migliorano in quasi tutti i casi quelli dei metodi esistenti in letteratura. Va tuttavia evidenziato come questi esiti siano dipendenti dai parametri utilizzati e non siano consistenti tra le varie collezioni e tra le varie lingue considerate. Non è stato inoltre individuato un set di parametri la cui efficacia sia indipendente dalla lingua in oggetto. Alcune fasi del trattamento del testo, in particolare quelle dipendenti dalla lingua, hanno incidenza elevata sugli *itemset* estratti e conseguentemente sul riassunto finale. Questo è in parte spiegabile dal fatto che gli strumenti di analisi specifici delle varie lingue possono produrre output diversi e anche dal fatto che esistono differenze intrinseche nelle lingue analizzate. I filtri specifici di ogni lingua utilizzano liste di termini e algoritmi di elaborazione che potrebbero essere oggetto di ulteriori approfondimenti.

Lo stesso vale per fasi indipendenti dalla lingua, come la generazione combinata di *n-gram* e *frequent itemset* che, prestandosi a varie modalità di implementazione, potrebbe essere realizzata in maniera più efficace rispetto alla versione attuale.

Un'altra fase di elaborazione con molti margini di miglioramento è la suddivisione del testo in frasi. In questo caso il *training* di nuovi modelli per il *sentence detection* garantirebbe una maggiore qualità dei riassunti.

L'utilizzo del metodo di Steinberger e Ježek per il calcolo del *salience score* permette di selezionare frasi significative che rischierebbero di essere ignorate da altri metodi ma al contempo aumenta la possibilità di selezionare frasi simili tra di loro. Il fenomeno diventa più evidente con l'aumentare della lunghezza del riassunto, come mostrato in appendice A. In questo senso potrebbero essere valutati dei correttivi nella funzione di calcolo del *salience score* e un meccanismo di selezione a posteriori delle frasi scelte per il riassunto, basato sulla similarità. Questo tipo di selezione avverrebbe sull'output del metodo, al contrario del *sentence pruning* già implementato e il cui ruolo è quello di rimuovere frasi quasi identiche anche nella formulazione e non solo simili nel contenuto.

Per il popolamento della matrice A è stato scelto uno schema binario ma potrebbero

essere valutati altri schemi come TF-IDF, TF-DF, LLR o Okapi BM25[1]

Molte combinazioni di parametri non sono state valutate perché, con la libreria attuale che utilizza solo matrici dense, avrebbero portato alla generazione di un numero molto elevato di *itemset*. L'utilizzo di una libreria che consenta la rappresentazione di matrici sparse e implementi la SVD parziale, permetterebbe quindi di eseguirne il calcolo.

[1] `https://www.elastic.co/guide/en/elasticsearch/guide/current/pluggable-similarites.html#bm25`

Appendici

Appendice A

Esempio di analisi del testo tramite la SVD

Per il supporto allo sviluppo del metodo e della relativa applicazione è risultato utile creare una funzione che consentisse di visualizzare l'output della SVD, rimappando gli indici sulle relative frasi e sui relativi *frequent itemset*. Nei listati che seguono sono rappresentate le quattro σ di valore più elevato, risultate dall'analisi di una delle collezioni del TAC201.

Per ognuno dei σ sono elencati:

- il valore di σ_i;
- gli itemset con indice più elevato per il sigma i-esimo, cioè gli elementi con valore più elevato della i-esima colonna di U;
- le frasi con indice più elevato, ovvero gli elementi con valore più elevato nella i-esima riga di V^T.

Le frasi sono rappresentate sia nella versione originale sia nella versione elaborata dai filtri (*tokenizer*, *lowercase*, *stopword removal*, *stemmer*).

Gli articoli all'interno della collezione sono incentrati sul tema del terremoto che ha colpito Haiti il 12 gennaio 2010. Si può notare come la distinzione dei topic non sia netta, con termini o sequenze di essi che appaiono tra i più rilevanti per diversi topic, ma allo stesso modo è evidente come per i vari σ siano diverse le frasi con indice più alto. Il σ_0 ad esempio sembra essere più incentrato sul tema del terremoto e della sua localizzazione mentre frasi e termini dei σ successivi spostano l'ambito verso il tema dei danni e successivamente degli aiuti umanitari.

```
*******************************************
Sigma 0 [6,77157739]
Top itemsets: {earthquak, haiti} [0,42137144], {au, princ} [0,32580838], {port, au, princ}
  [0,32580838], {magnitud, earthquak} [0,30716615], {magnitud, earthquak, haiti} [0,28172944], {7.0,
  magnitud, earthquak} [0,24362368], {7.0, magnitud, earthquak, haiti} [0,23052774], {earthquak,
  peopl} [0,20249847], {earthquak, peopl, haiti} [0,17322356], {capit, port, princ, au} [0,14755424],
   {nation, haiti} [0,12356201], {12, januari} [0,12160565], {12, januari, left, peopl} [0,11888873],
   {nation, earthquak, haiti} [0,11168858], {port, au, princ, haiti} [0,11147528],

Top Sentences:
[0,47446652] An earthquake, with a magnitude of 7.0, struck Haiti on January 12, killing as much as
  200,000 people and largely destroying the capital Port-au-Prince; another million have been left
  without homes.
#194    [earthquak, magnitud, 7.0, struck, haiti, januari, 12, kill, 200,000, peopl, destroi, capit,
  port, au, princ, left, home]

[0,31134794] A strong earthquake has once again struck Haiti, eight days after a 7.0 magnitude quake
  left much of the nation in ruins.
#38     [strong, earthquak, struck, haiti, dai, 7.0, magnitud, quak, left, nation, ruin]

[0,28176639] A massive earthquake, registering 7.0 on the moment magnitude scale, struck Haiti
  yesterday, destroying many buildings, disrupting communications, and burying an unknown number of
  people underneath rubble.
#50     [massiv, earthquak, regist, 7.0, moment, magnitud, scale, struck, haiti, yesterdai, destroi,
  build, disrupt, commun, buri, unknown, number, peopl, underneath, rubbl]

[0,28067876] As reported the United Nations, January 12 Haiti earthquake left exactly 222,570 deaths,
   1,300,000 refugees in harbours, 766,000 displaced people, 310,000 injured and 869 disappeared.
#246    [report, unit, nation, januari, 12, haiti, earthquak, left, exactli, 222,570, death,
  1,300,000, refuge, harbour, 766,000, displac, peopl, 310,000, injur, 869, disappear]


*******************************************
Sigma 1 [5,89093331]
Top itemsets: {au, princ} [0,52997513], {port, au, princ} [0,52997513], {earthquak, haiti}
  [0,29019886], {port, au, princ, report} [0,21760918], {magnitud, earthquak} [0,19543805], {hospit,
  port, princ, au} [0,18156048], {magnitud, earthquak, haiti} [0,17483117], {capit, port, princ, au}
  [0,14759222], {hospit, port, princ, au, report} [0,14381946], {7.0, magnitud, earthquak}
  [0,14076891], {7.0, magnitud, earthquak, haiti} [0,12985178], {nation, haiti} [0,10712287], {nation
  , earthquak, haiti} [0,09711792], {earthquak, peopl} [0,09685882], {unit, nation} [0,08318880],

Top Sentences:
[0,29258936] "We have reports of some of the most important hospitals in Port-au-Prince have been
  severely impacted by the earthquake," said Paul Conneally, the Head of Media for the International
  Federation of Red Cross and Red Crescent Societies.
#69     [report, hospit, port, au, princ, sever, impact, earthquak, paul, conneal, head, media,
  intern, feder, red, cross, red, crescent, societi]

[0,28041287] No deaths have yet been reported, but a hospital in Port-au-Prince was damaged, and a US
   government official said several houses fell into a ravine.
#15     [death, report, hospit, port, au, princ, damag, govern, offici, hous, fell, ravin]

[0,27422863] Hospitals in Port-au-Prince were reported to have collapsed, raising fears that the
  injured would not be able to receive treatment easily.
#68     [hospit, port, au, princ, report, collaps, rais, fear, injur, receiv, treatment, easili]

[0,24306742] A strong earthquake has once again struck Haiti, eight days after a 7.0 magnitude quake
  left much of the nation in ruins.
#38     [strong, earthquak, struck, haiti, dai, 7.0, magnitud, quak, left, nation, ruin]
```

```
*********************************************
Sigma 2 [5,20013841]
Top itemsets: {unit, state} [0,35348115], {state, haitian} [0,32276106], {unit, nation} [0,29137906],
   {unit, haiti} [0,26442207], {govern, unit} [0,21961392], {nation, haitian} [0,18156603], {state,
  children} [0,17924986], {child, haiti} [0,17242342], {countri, haiti} [0,17169540], {countri,
  haitian} [0,17095867], {countri, state} [0,16843786], {state, haiti} [0,16442736], {magnitud,
  earthquak} [0,16028448], {ten, haiti} [0,15174380], {child, countri, haiti} [0,14888234],

Top Sentences:
[0,58018789] Ten United States missionaries who tried to take 33 Haitian children out the country
  last week without the 'governments consent have been charged with child kidnapping and criminal
  association for illegally trying to take children out of Haiti.
#237     [ten, unit, state, missionari, 33, haitian, children, countri, week, govern, consent, charg,
  child, kidnap, crimin, associ, illeg, children, haiti]

[0,30391096] Haitian police yesterday arrested ten United States nationals, five men and five women,
  over the alleged abduction of 33 children.
#208     [haitian, polic, yesterdai, arrest, ten, unit, state, nation, men, women, alleg, abduct, 33,
  children]

[0,30039401] The Haitian ambassador to the United States, Raymond Joseph, told CNN the Caribbean
  nation is seeking US assistance, and called the quake a catastrophe of major proportions.
#98     [haitian, ambassador, unit, state, raymond, joseph, told, cnn, caribbean, nation, seek,
  assist, call, quak, catastroph, major, proport]

[0,21985837] The United Nations has announced that the government of Haiti has put an end to its
  efforts to find and rescue buried survivors of the earthquake that hit the region eleven days ago.
#125     [unit, nation, announc, govern, haiti, effort, find, rescu, buri, survivor, earthquak, hit,
  region, eleven, dai, ago]


*********************************************
Sigma 3 [4,54549232]
Top itemsets: {unit, nation} [0,29659395], {countri, haiti} [0,27727220], {unit, nation, haiti}
  [0,24708598], {nation, haiti} [0,22585565], {magnitud, earthquak} [0,20793054], {magnitud,
  earthquak, haiti} [0,19004765], {child, haiti} [0,17866970], {report, earthquak} [0,17131632], {
  report, haiti} [0,16976972], {nation, earthquak, haiti} [0,16882786], {report, peopl} [0,16810528],
   {7.0, magnitud, earthquak} [0,16446243], {nation, report} [0,15901683], {injur, report}
  [0,15883482], {state, haitian} [0,15362899],

Top Sentences:
[0,57842053] As reported the United Nations, January 12 Haiti earthquake left exactly 222,570 deaths,
   1,300,000 refugees in harbours, 766,000 displaced people, 310,000 injured and 869 disappeared.
#246     [report, unit, nation, januari, 12, haiti, earthquak, left, exactli, 222,570, death,
  1,300,000, refuge, harbour, 766,000, displac, peopl, 310,000, injur, 869, disappear]

[0,38848589] Ten United States missionaries who tried to take 33 Haitian children out the country
  last week without the 'governments consent have been charged with child kidnapping and criminal
  association for illegally trying to take children out of Haiti.
#237     [ten, unit, state, missionari, 33, haitian, children, countri, week, govern, consent, charg,
  child, kidnap, crimin, associ, illeg, children, haiti]

[0,28548756] The United Nations has announced that the government of Haiti has put an end to its
  efforts to find and rescue buried survivors of the earthquake that hit the region eleven days ago.
#125     [unit, nation, announc, govern, haiti, effort, find, rescu, buri, survivor, earthquak, hit,
  region, eleven, dai, ago]

[0,25921935] Buildings across the capital have collapsed, including the presidential palace and the
  headquarters of the United Nations peacekeeping mission in Haiti.
#56     [build, capit, collaps, includ, presidenti, palac, headquart, unit, nation, peacekeep,
  mission, haiti]
```

Di seguito viene riportato un riassunto automatico delle collezione di news sul terremoto di Haiti generato da LSA-itemset summarizer . I temi principali sono riportati ma sono presenti alcune frasi ridondanti:

```
An earthquake, with a magnitude of 7.0, struck Haiti on January 12, killing as much as 200,000 people
 and largely destroying the capital Port-au-Prince; another million have been left without homes.
Ten United States missionaries who tried to take 33 Haitian children out the country last week
without the 'governments consent have been charged with child kidnapping and criminal association for
 illegally trying to take children out of Haiti.
As reported the United Nations, January 12 Haiti earthquake left exactly 222,570 deaths, 1,300,000
refugees in harbours, 766,000 displaced people, 310,000 injured and 869 disappeared.
The United Nations has announced that the government of Haiti has put an end to its efforts to find
and rescue buried survivors of the earthquake that hit the region eleven days ago.
A massive earthquake, registering 7.0 on the moment magnitude scale, struck Haiti yesterday,
destroying many buildings, disrupting communications, and burying an unknown number of people
underneath rubble.
A strong earthquake has once again struck Haiti, eight days after a 7.0 magnitude quake left much of
the nation in ruins.
The survivors from the recent 7.0 magnitude earthquake in Haiti are now facing increasing insecurity
from human traffickers and convicts escaped from collapsed prisons, officials have cautioned, even as
 aid is flowing into the country.
```

Bibliografia

[1] Peter Abeles. *Efficient Java Matrix Library*. URL: http://ejml.org.

[2] Rakesh Agrawal, Tomasz Imielinski e Arun Swami. «Mining Association in Large Databases». In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93* (1993), pp. 207–216. DOI: 10.1145/170036.170072.

[3] Apache Software Foundation. *Apache Lucene*. URL: https://lucene.apache.org/.

[4] Apache Software Foundation. *Apache OpenNLP*. URL: http://opennlp.apache.org/.

[5] Elena Baralis et al. «Multi-document summarization exploiting frequent itemsets». In: *Proceedings of the 27th Annual …* (2012), pp. 782–786. DOI: 10.1145/2245276.2245427.

[6] Elena Baralis et al. «MWI-Sum: A Multilingual Summarizer Based on Frequent Weighted Itemsets». In: *ACM Transactions on Information Systems* 34.1 (2015), pp. 1–35. DOI: 10.1145/2809786.

[7] Regina Barzilay e Michael Elhadad. «Using lexical chains for text summarization». In: *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization* 17.48 (1997), pp. 10–17. DOI: 10.3115/1034678.1034760. arXiv: arXiv:1011.1669v3.

[8] Jill Burstein e Daniel Marcu. «Toward Using Text Summarization for Essay-Based Feedback». In: *Proceedings of TALN 2000, Conference Lausanne, Switzerland, 2000* (2000), pp. 16–18.

[9] Jaime Carbonell e Jade Goldstein. «The use of MMR, diversity-based reranking for reordering documents and producing summaries». In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '98* (1998), pp. 335–336. DOI: 10.1145/290941.291025.

[10] Ht Dang e Karolina Owczarzak. «Overview of the TAC 2008 update summarization task». In: *Tac* (2008), pp. 1–16.

[11] U.S. Department of Defense. *DEPARTMENT OF DEFENSE FY 2006/FY 2007 Budget Estimates*. Rapp. tecn. 2005, p. 29.

[12] Ted Dunning. «Accurate Methods for the Statistics of Surprise and Coincidence». In: *Computational Linguistics* 19 (1993), pp. 61–74.

[13] Harold P. Edmundson. «New Methods in Automatic Extracting». In: *Journal of the ACM* 16.2 (1969), pp. 264–285. DOI: 10.1145/321510.321519.

[14] Güneş Erkan e Dragomir R. Radev. «LexRank: Graph-based lexical centrality as salience in text summarization». In: *Journal of Artificial Intelligence Research* 22 (2004), pp. 457–479. DOI: 10.1613/jair.1523. arXiv: 1109.2128.

[15] George W. Furnas et al. «Information retrieval using a singular value decomposition model of latent semantic structure». In: *11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1988)* (1988), pp. 465–480. DOI: 10.1145/62437.62487.

[16] George Giannakopoulos e Vangelis Karkaletsis. «Summarization system evaluation variations based on n-gram graphs». In: (2010).

[17] George Giannakopoulos et al. «Summarization System Evaluation Revisited: N-Gram Graphs». In: *ACM Transactions on Speech and Language Processing* 5.3 (2008), pp. 1–39. DOI: 10.1145/1410358.1410359.

[18] George Giannakopoulos et al. «TAC 2011 MultiLing Pilot Overview». In: November (2011).

[19] Dan Gillick e Yang Liu. «Non-Expert Evaluation of Summarization Systems is Risky». In: *Proceedings NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* 2009.June (2010), pp. 148–151.

[20] Gene H Golub e Charles F Van Loan. *Matrix Computations*. 2013, p. 780.

[21] Yihong Gong e Xin Liu. «Generic Text Summarization Using Relevance Measure and Latent Semantic Snalysis». In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01* (2001), pp. 19–25. DOI: 10.1145/383952.383955.

[22] Vasileios Hatzivassiloglou et al. «SIMFINDER : A Flexible Clustering Tool for Summarization». In: *Proceedings of the NAACL Workshop on Automatic Summarization*. 2001, pp. 41–49.

[23] Jiri Hynek e Karel Ježek. «Practical approach to automatic text summarization». In: *Proceedings of the ELPUB* (2003).

[24] K.S. Jones et al. «Automatic summarizing: factors and directions». In: *Advances in automatic text summarization* (1999), pp. 1–12. DOI: 10.1145/375551.375604. arXiv: 9805011v1 [arXiv:cmp-lg].

[25] Jon M. Kleinberg. «Authoritative sources in a hyperlinked environment». In: *Journal of the ACM* 46.5 (1999), pp. 604–632. DOI: 10.1145/324133.324140. arXiv: 0208024 [gr-qc].

[26] Julian Kupiec, Jan Pedersen e Francine Chen. «A trainable document summarizer». In: *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '95*. 1995, pp. 68–73. DOI: 10.1145/215206.215333.

[27] Thomas K Landauer e Susan T. Dumais. «A solution to Plato ' s problem : The Latent Semantic Analysis Theory of Acquisition , Induction , and Representation of Knowledge». In: *Psychological Review* 104.2 (1997), pp. 211–240. DOI: 10.1037/0033-295X.104.2.211.

[28] Thomas K Landauer, Peter W Folt e Darrell Laham. «An introduction to latent semantic analysis». In: *Discourse processes* 25.2 (1998), pp. 259–284. DOI: 10.1080/01638539809545028.

[29] Chin-Yew Lin. «Rouge: A package for automatic evaluation of summaries». In: *Proceedings of the workshop on text summarization branches out (WAS 2004)* 1 (2004), pp. 25–26.

[30] Chin-Yew Lin e Eduard Hovy. «The automated acquisition of topic signatures for text summarization». In: *Proceedings of the 18th conference on Computational linguistics* 1 (2000), pp. 495–501. DOI: 10.3115/990820.990892.

[31] Annie Louis e Ani Nenkova. «Automatic Summary Evaluation without Human Models». In: *Analysis* (2008).

[32] Annie Louis e Ani Nenkova. «Automatically Assessing Machine Summary Content Without a Gold Standard». In: *Computational Linguistics* 39.2 (2013), pp. 267–300. DOI: 10.1162/COLI_a_00123. arXiv: 1309.4408.

[33] H. P. Luhn. «The Automatic Creation of Literature Abstracts». In: *IBM Journal of Research and Development* 2.2 (1958), pp. 159–165. DOI: 10.1147/rd.22.0159.

[34] Michael Mampaey, Nikolaj Tatti e Jilles Vreeken. «Tell Me What I Need to Know: Succinctly Summarizing Data With Itemsets». In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11* (2011), pp. 573–581. DOI: 10.1145/2020408.2020499.

[35] Manuel J Maña-López, Manuel De Buenaga e José María Gómez Hidalgo. «Multidocument summarization: An added value to clustering in interactive retrieval». In: *ACM Transactions on …* 22.2 (2004), pp. 215–241. DOI: 10.1145/984321.984323.

[36] Inderjeet Mani et al. «SUMMAC: a text summarization evaluation». In: *Natural Language Engineering* 8.01 (2002), pp. 43–68. DOI: 10.1017/S1351324901002741.

[37] Kathleen R Mckeown et al. «Tracking and summarizing news on a daily basis with Columbia's Newsblaster». In: *Proceedings of the Human Language Technology Conference. San Diego, Ca.* (2002), pp. 280–285. DOI: 10.3115/1289189.1289212.

[38] Kathleen Mckeown et al. «Do Summaries Help? A Task-Based Evaluation of Multi-Document Summarization». In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 210–217. DOI: 10.1145/1076034.1076072.

[39] Rada Mihalcea e Paul Tarau. «TextRank: Bringing order into texts». In: *Proceedings of EMNLP* 85 (2004), pp. 404–411. DOI: 10.3115/1219044.1219064. arXiv: arXiv:1011.1669v3.

[40] George A. Miller et al. «Introduction to wordnet: An on-line lexical database». In: *International Journal of Lexicography* 3.4 (1990), pp. 235–244. DOI: 10.1093/ijl/3.4.235.

[41] Andrea Morandi et al. «X-ray, lensing and Sunyaev-Zel'dovich triaxial analysis of Abell 1835 out to R200». In: *Monthly Notices of the Royal Astronomical Society* 425.3 (set. 2012), pp. 2069–2082. DOI: 10.1111/j.1365-2966.2012.21196.x. arXiv: 1111.6189.

[42] G Murray, S Renals e J Carletta. «Extractive Summarization of Meeting Recordings». In: *Proceedings Interspeech* (2005), pp. 593–596.

[43] Ani Nenkova e Kathleen McKeown. «Automatic Summarization». In: *Foundations and Trends® in Information Retrieval* 5.3 (2011), pp. 235–422. DOI: 10.1561/1500000015.

[44] NIST. *Document Understanding Conferences - DUC 2004*. 2004. URL: http://duc.nist.gov/pubs.html%7B%5C#%7D2004.

[45] NIST. *TAC 2011 Summarization Track*. 2011. URL: https://tac.nist.gov/2011/Summarization/.

[46] Paul Over e James Yen. *An Introduction to DUC-2004 Intrinsic Evaluation of Generic News Text*. 2004.

[47] Makbule Gulcin Ozsoy, IIlyas Cicekli e Ferda Nur Alpaslan. «Text summarization of turkish texts using latent semantic analysis». In: *Proceedings of the 23rd international conference on computational linguistics* August (2010), pp. 869–876.

[48] C D Paice. «The Automatic Generation of Literature Abstracts: An Approach Based on the Identification of Self-indicating Phrases». In: *Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval*. 1981, pp. 172–191.

[49] Chris D. Paice. «Constructing literature abstracts by computer: Techniques and prospects». In: *Information Processing and Management* 26.1 (1990), pp. 171–186. DOI: 10.1016/0306-4573(90)90014-S.

[50] Dmitri G. Roussinov e Hsinchun Chen. «Information navigation on the web by clustering and summarizing query results». In: *Information Processing and Management* 37.6 (2001), pp. 789–816. DOI: 10.1016/S0306-4573(00)00062-5.

[51] Karen Sparck Jones. «A Statistical Interpretation of Term Specificity and its Retrieval». In: *Journal of Documentation* 28.1 (1972), pp. 11–21. DOI: 10.1108/eb026526. arXiv: arXiv:1011.1669v3.

[52] Josef Steinberger e Karel Ježek. «Using Latent Semantic Analysis in Text Summarization». In: *In Proceedings of ISIM 2004* (2004), pp. 93–100.

[53] G. W. Stewart. «On the Early History of the Singular Value Decomposition». In: *SIAM Review* 35 (1993), pp. 551–566. DOI: 10.1137/1035134. arXiv: arXiv:1011.1669v3.

[54] Simone Teufel. «Task-based evaluation of summary quality: Describing relationships between scientific papers». In: *In Workshop Automatic Summarization, NAACL* 102 (2001), pp. 12–21.

[55] Khushboo S. Thakkar, R. V. Dharaskar e M. B. Chandak. «Graph-based algorithms for Text Summarization». In: *Proceedings - 3rd International Conference on Emerging Trends in Engineering and Technology, ICETET 2010* Vi (2010), pp. 516–519. DOI: 10.1109/ICETET.2010.104.

[56] Anastasios Tombros e Mark Sanderson. «Advantages of query biased summaries in information retrieval». In: *Proceedings of the 1998 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)* (1998), pp. 2–10. DOI: 10.1145/290941.290947.

[57] Takeaki Uno, Masashi Kiyomi e Hiroki Arimura. «LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets». In: *Workshop on Frequent Itemset Mining* (2004).

[58] Xiaojun Wan e Jianwu Yang. «Improved Affinity Graph Based Multi-Document Summarization». In: *Human Language Technology Conference of the North American Chapter of the ACL* June (2006), pp. 181–184.

[59] Dingding Wang e Multi-document Summarization. «Document Update Summarization Using Incremental Hierarchical Clustering CLUSTERING BASED DOCUMENT». In: *Update* (2010), pp. 279–287. DOI: 10.1145/1871437.1871476.

[60] Wei Wang, Jiong Yang e Philip S. Yu. «Efficient mining of weighted association rules (WAR)». In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00* (2000), pp. 270–274. DOI: 10.1145/347090.347149.

[61] Zi Yang et al. «Social context summarization». In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (2011), pp. 255–264. DOI: 10.1145/2009916.2009954.