ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Dipartimento di Informatica – Scienza e Ingegneria

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

TESI DI LAUREA

in

Tecnologie Web T

STIME DI SOSTENIBILITÀ AZIENDALE ATTRAVERSO STRUMENTI WEB BASATI SU DATI SECONDARI

CANDIDATO

Andrea Saiani

RELATORE

Chiar.mo Prof. Paolo Bellavista

CORRELATORI

Dott. Ing. Luca Foschini

Dott. Ing. Matteo Mura

Anno Accademico 2015/2016

Sessione II

Sommario

Introduzione 5

Capitolo 1 Osservatorio per la sostenibilità aziendale 7

1.1 La sostenibilità aziendale 7

1.2 Green Economy 8

1.2.1 Definizione 8

1.2.2 Green Economy in Italia e in Emilia-Romagna 9

1.3 Osservatorio 10

1.3.1 Creazione e popolamento del database 10

1.3.2 Analisi dei processi e degli indicatori 10

1.3.3 Popolamento del database 11

Capitolo 2 Problema 13

2.1 Descrizione del problema 13

2.1.1 Obiettivi 14

2.2 Analisi del problema 14

2.2.1 Gestione degli input 14

2.2.2 Download e lettura dei siti web 18

2.2.3 Processing dei dati 22

2.2.4 Visualizzazione e confronto dei risultati 23

Capitolo 3 Strumenti utilizzati 25

3.1 Software 25

3.1.1 GNU Wget 25

3.2 Librerie 27

3.2.1 Jsoup 28

3.2.2 OpenCsv 29

Capitolo 4 Implementazione e confronto 31
4.1 Progettazione e realizzazione 31
4.1.1 Lettura degli input 32
4.1.2 Download e parsing dei siti web 37
4.1.3 Ricerca dei KPI 41
4.1.4 Visualizzazione dei risultati 43
4.2 Confronto dei risultati 48
4.3 Analisi delle criticità 53
Capitolo 5 Riprogettazione, nuovo confronto e analisi delle performance 55
5.1 Analisi delle modifiche 55
5.2 Implementazione delle modifiche 56
5.3 Visualizzazione dei risultati 62
5.4 Confronto finale 66
5.5 Analisi delle performance 70
Conclusioni 75
Bibliografia e Sitografia 77

Introduzione

L'attività di ricerca e analisi dei dati è una di quelle operazioni che ancora non sfruttano a pieno le potenzialità dell'informatica.
Un esempio è quello di alcuni colleghi di Ingegneria Gestionale che hanno svolto un'analisi riguardo la sostenibilità aziendale. L'attività svolta consiste nella ricerca di alcuni indicatori, da loro definiti, all'interno dei siti web di certe aziende, i quali però possono essere composti anche da migliaia di pagine ciascuno. Il lavoro risulta essere perciò molto dispendioso in termini di tempo e persone richieste.
L'oggetto di questa tesi è quello di creare strumenti che consentano di automatizzare questa attività di analisi per effettuare stime di sostenibilità aziendale basate su dati secondari. Verrà sviluppato un software in grado di accedere ai siti web delle aziende e di cercare specifici indicatori all'interno delle pagine.
Dopo un'analisi iniziale del problema, verranno introdotti gli strumenti esterni che saranno necessari per realizzare l'implementazione e infine verrà fatto un confronto con le misurazioni reali.

La presente tesi si divide in cinque capitoli. Il Capitolo 1 introduce l'osservatorio e alcuni concetti legati alla sostenibilità, per inquadrare meglio il dominio a cui è applicato il problema.

Nel Capitolo 2 viene effettuata un'analisi dettagliata del problema, partendo prima dalla comprensione dei dati che sono forniti input, poi cercando di trovare possibili soluzioni al problema. Inoltre verranno fissati gli obiettivi.

Il Capitolo 3 descrive gli strumenti che verranno utilizzati, ovvero software e librerie che serviranno a migliorare e semplificare il lavoro.

L'implementazione del software è dettagliata nel Capitolo 4, accompagnata da un confronto con le rilevazioni effettuate a mano e da eventuali miglioramenti che possono essere apportati.

Il Capitolo 5 riprogetta l'algoritmo ed esegue un nuovo confronto prendendo un campione più ampio di aziende quindi analizza le performance del software.

Capitolo 1 Osservatorio per la sostenibilità aziendale

1.1 La sostenibilità aziendale

Il concetto di sostenibilità trova le proprie origini in ambito di studi ecologici e rimanda al "potenziale di un ecosistema di sussistere nel tempo, senza alcun cambiamento" [Jabareen, 2008], anche se la tematica della sostenibilità richiama immediatamente in causa differenti campi del sapere: ambientali, economici e sociali.

La sostenibilità ambientale ha come obiettivo la conservazione dell'ecosistema, con particolare attenzione ai "processi biologici naturali" ed alla sua "costante produttività e funzionamento" [Beckermann, 1994].

La sostenibilità economica significa che "il capitale non dovrebbe decrescere per non mettere in pericolo le possibilità delle generazioni future di generare ricchezza e benessere" [Jabareen, 2008]. Affinché un atto si possa definire economicamente sostenibile, si richiede che i benefici superino i costi, o quanto meno li eguaglino.

La sostenibilità sociale garantisce condizioni di benessere umano (sicurezza, salute, istruzione) equamente distribuite per classi e genere.

In **Figura 1.1** viene mostrato quanto appena detto.

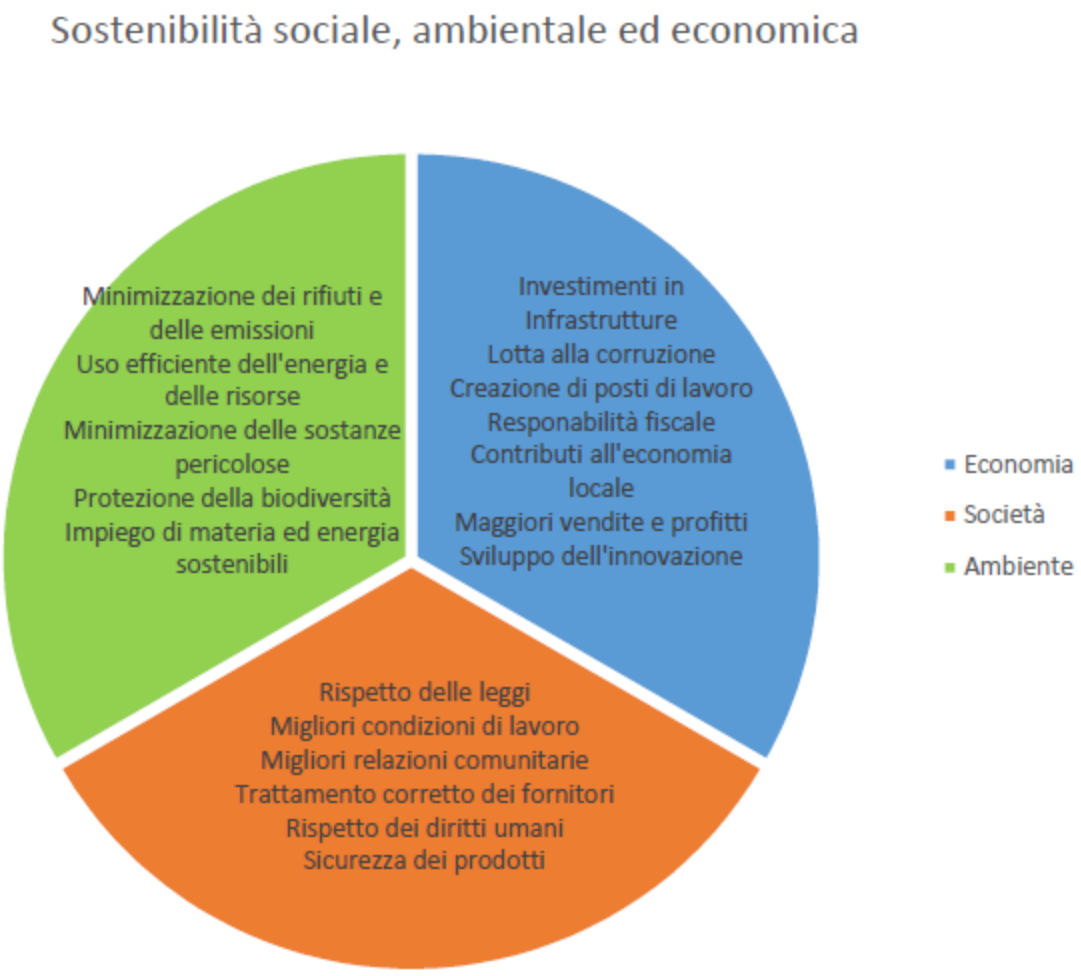

Figura 1.1: Schema operativo di impresa sostenibile dell'oced, 2011

1.2 Green Economy

1.2.1 Definizione

Con il termine Green Economy si intende un modello "capace di produrre un benessere di migliore qualità e più equamente esteso, migliorando la qualità dell'ambiente e salvaguardando il capitale naturale" [UNEP, 2009].

In generale con questo modello si dovrebbero creare investimenti al fine di ridurre le emissioni di anidride carbonica, migliorare l'efficienza energetica delle risorse e prevenire la scomparsa della biodiversità.
Per raggiungere questi obiettivi è necessario lavorare sui processi: un processo si definisce "green" quando, a parità di output produttivo, è in grado di ridurre l'impiego di materia prima, di energia, di impatto ambientale (atmosfera, acqua, suolo), di rifiuti non riutilizzabili.

1.2.2 Green Economy in Italia e in Emilia-Romagna

L'Italia si conferma leader nell'utilizzo di strumenti volontari per la gestione della sostenibilità, cioè di tutte quelle certificazioni ambientali e sociali che permettono ad un'azienda di dimostrare la propria efficienza e attenzione all'ambiente e ai lavoratori.
In particolare tra le certificazioni più importanti e utilizzate troviamo:

- ISO14001: Fornisce strumenti pratici per aziende e organizzazioni di tutti i tipi che vogliono gestire le proprie responsabilità sociali (etichettatura, comunicazione, analisi del ciclo di vita, sfide ambientali come il cambiamento climatico);
- EMAS: è uno strumento di gestione sviluppato dalla Commissione europea per le aziende e altre organizzazioni al fine di valutare, fare report e migliorare le proprie prestazioni ambientali;
- FSC: il marchio FSC identifica i prodotti contenenti legno proveniente da foreste gestite in maniera corretta e responsabile secondo rigorosi standard ambientali, sociali ed economici;
- OHSAS18001: assicura l'ottemperanza ai requisiti previsti per i Sistemi di Gestione della Salute e Sicurezza sul Lavoro e consente a un'Organizzazione di valutare meglio i rischi e migliorare le proprie prestazioni;
- SA8000: è uno standard internazionale che elenca i requisiti per un comportamento eticamente corretto delle imprese e della filiera di produzione verso i lavoratori (diritti umani e dei lavoratori, sicurezza sul lavoro).

1.3 Osservatorio

L'Osservatorio delle aziende che applicano sostenibilità nei propri processi in Emilia-Romagna nasce con l'intenzione di dare, a chiunque sia interessato nella situazione socio-ambientale delle imprese di questa regione, uno strumento che sia in grado di mappare e quantificare in maniera standard e affidabile la presenza (o l'assenza) di determinate certificazioni, politiche o attenzioni delle aziende verso la sostenibilità.

1.3.1 Creazione e popolamento del database

Una volta definiti i parametri per la scelta delle aziende da analizzare è necessario creare un database delle informazioni che si ritengono importante al fine di quantificare i processi di sostenibilità. Sono stati individuati 11 Processi e 63 Indicatori [Mandrioli, 2016].
Un indicatore chiave di prestazione (in inglese Key Performance Indicator o KPI) è un indice che monitora l'andamento di un processo aziendale.

1.3.2 Analisi dei processi e degli indicatori

I Processi non sono altro che il raggruppamento di Indicatori sulla base di un tema di sostenibilità comune. In particolare:

0) Informazioni Generali: Nome, Partita Iva, Codice Ateco, Fatturato, Dipendenti, Sito Web, Ubicazione;
1) Certificazioni Ambientali: ISO14001, EMAS, ISO50001, Etichetta ECOLABEL, LCA, FSC, GOLDPOWER, LEED;
2) Certificazioni Sociali: SA8000, ISO26000, OHSAS18001, IFS, ISO22005, ISO22000;
3) Energia: presenza di impianti di energia rinnovabile, politiche di risparmio energetico, indicazione di costi e risparmi, report energetico, azioni future, politiche di sensibilizzazione in materia energetica;
4) Risorse Primarie: presenza di impianti di trattamento/depurazione/captazione di acque reflue o piovane, indicazione livello di riutilizzo di acqua, report idrico, azioni future, politiche di sensibilizzazione in materia idrica;

5) Gestione Rifiuti: riferimento a normative, raccolta differenziata e modalità, riutilizzo dei rifiuti per produzione di riscaldamento/elettricità/nuovi prodotti, riciclo dei propri prodotti in fase di smaltimento, packaging biodegradabile e riutilizzabile, report sui rifiuti, azioni future, politiche di sensibilizzazione in materia di riutilizzo e riciclo;
6) Impatto Ambientale: monitoraggio delle emissioni in aria/acqua/terra;
7) Reporting: presenza di bilancio di sostenibilità, azioni future, approfondimenti tematici;
8) Welfare: presenza di politica pari opportunità, asili nido, flessibilità oraria, assistenza sanitaria;
9) Responsabilità Sociale (CSR): presenza di codice etico, politiche di formazione dipendenti, valutazione impatto ambientale, analisi del rischio e sicurezza sul lavoro;
10) Supply Chain: presenza di criteri ambientali e sociali di selezione dei fornitori;
11) Valore per il Consumatore: presenza di politiche di incentivazione per la restituzione di un prodotto vecchio, trasparenza del prodotto, politiche di comunicazione sulla sostenibilità del prodotto.

1.3.3 Popolamento del database

Il popolamento del database è stato effettuato per ciascuna azienda mediante un'analisi diretta sulla pagina web o su articoli. Nel caso venisse riscontrata la presenza di uno degli indicatori precedentemente definiti è stata registrata con un 1, altrimenti l'assenza con un 0.

Capitolo 2 Problema

2.1 Descrizione del problema

Il dominio trattato è quello della sostenibilità aziendale applicato ad alcune aziende dell'Emilia-Romagna. Nel nostro caso specifico le aziende trattate appartengono tutte al medesimo settore merceologico, ovvero quello della fabbricazione di macchinari ed apparecchiature nca (codice ateco 28). Tutte le aziende sono state sottoposte alla stessa analisi, in modo da creare uno strumento che non sia legato solo ad alcuni casi particolari ma utilizzabile indipendentemente dal dominio a cui è applicato. Inoltre è stata fatta una classificazione basata sulle dimensioni, tenendo conto del fatturato e del numero dei dipendenti.

L'idea di base è quella di realizzare un software in grado di rendere automatico il processo di ricerca dei KPI all'interno dei siti web, e quindi utilizzabile per analizzare velocemente un gran numero di aziende. In particolare il software dovrà essere in grado, prendendo come input una lista di aziende e un elenco di KPI d'interesse, di effettuare la ricerca di quest'ultimi nei rispettivi siti web delle aziende e di mostrarne in output i risultati. Inoltre dovrà mostrare il confronto con le misurazioni reali.

Dal momento che i processi e i KPI sono stati definiti da altri, la misurazione sarà basata su dati secondari. I dati primari sono quelli raccolti attraverso attività in prima persona, come osservazione, registrazione, misurazione, ispezione di persone (e loro comportamenti), oggetti o eventi. I dati secondari sono invece quelli già raccolti da altre persone, organizzazioni o amministrazioni, come nel caso di vari tipi di documenti, diari, statistiche ufficiali e altre ricerche.

2.1.1 Obiettivi

Trattandosi di un primo approccio alla risoluzione di questo genere di problema, un primo obiettivo è sicuramento quello di realizzare un software in grado di automatizzare l'operazione di ricerca degli indicatori all'interno di siti web, cercando di ottenere un grado di precisione quanto più vicino alle rilevazioni già effettuate a mano.
Un altro obiettivo è quello di creare un software flessibile e riutilizzabile, non legato al dominio specifico di questa tesi ma potenzialmente in grado di essere sfruttabile in ambiti diversi, ad esempio per analizzare altre categorie di aziende oppure utilizzando processi e indicatori diversi, modificando quindi i parametri della ricerca stessa.
In secondo piano è interessante anche l'aspetto legato alle performance: si cercherà di creare un software veloce e ottimizzato, data la mole di dati da processare, che sia in grado di compiere il lavoro in un tempo ragionevole e senza richiedere troppe risorse.

2.2 Analisi del problema

Il problema si divide in quattro macro fasi:

- Gestione degli input
- Download e lettura dei siti web
- Processing dei dati
- Visualizzazione e confronto dei risultati

In ogni sottosezione verrà analizzata singolarmente ciascuna parte.

2.2.1 Gestione degli input

La prima fase riguarda gli input. Sono dati tre file: uno contenente la lista di aziende, un altro con l'elenco dei processi e relativi KPI e infine quello che contiene le misurazioni reali.
Tutti e tre sono file Excel, in quanto contengono le informazioni in forma tabellare. Per leggerne contenuto sarà necessario innanzitutto convertirli in un formato più adeguato, ovvero il **CSV**.

Il comma-separated values (abbreviato in CSV) è un formato di file basato su file di testo utilizzato per l'importazione ed esportazione (ad esempio da fogli elettronici o database) di una tabella di dati. Non esiste uno standard formale che lo definisca, ma solo alcune prassi più o meno consolidate.
In questo formato, ogni riga della tabella (o record della base dati) è normalmente rappresentata da una linea di testo, che a sua volta è divisa in campi (le singole colonne) separati da un apposito carattere separatore, ciascuno dei quali rappresenta un valore.
Il formato CSV non specifica una codifica di caratteri, né la convenzione per indicare il fine linea (nei sistemi operativi Unix viene usato il carattere ASCII line-feed, nei sistemi operativi Microsoft Windows si usa la sequenza di caratteri ASCII carriage return+line-feed, mentre in altri può essere usato il solo carattere carriage return), né il carattere da usare come separatore tra campi e nemmeno convenzioni per rappresentare date o numeri (tutti i valori sono considerati come semplici stringhe di testo) e se la prima riga è solo di intestazione o meno. Questi dettagli possono dover essere specificati dall'utente tutte le volte che si importano o esportano dati in formato CSV in un programma come ad esempio un foglio elettronico.

Per leggere i file è necessario l'utilizzo di un **parser**: si tratta di un programma che analizza un flusso di dati in ingresso, verificandone la correttezza sintattica rispetto ad una data grammatica. L'ingresso ad un parser è generalmente testo in linguaggio informatico, ma può anche essere testo in un linguaggio naturale. Il parser generalmente contiene anche un analizzatore lessicale, il quale crea una serie di token dalla sequenza di caratteri di input e ne controlla la correttezza. Col termine parsing si indica in generale l'insieme dell'analisi lessicale con quella sintattica vera e propria.
Servirà quindi un parser specifico per la grammatica CSV: una volta estratto il contenuto, questo sarà memorizzato nel sistema tramite strutture adeguate per ciascun file.

1) File aziende

In **Figura 2.1** viene mostrato il file contenente le aziende.

	COD. ATECO 28 FABBRICAZIONE DI MACCHINARI ED APPARECCHIATURE NCA					
			INFO			
	1	2	3	4	5	6
n.	NOME AZIENDA	P .IVA	CODICE ATECO	FATTURATO	N. DIPENDENTI	SITO WEB
1	SACMI COOPERATIVA MECCANICI	IT00498321207	289999	843.172.000	1085	http://www.sacmi.it/
2	BONFIGLIOLI RIDUTTORI S.P.A.	IT04984850968	281510	430.135.000	1240	http://www.bonfiglioli.it/it-it/
3	TETRA PAK PACKAGING SOLU	IT00907680367	282930	389.289.000	838	http://www.tetrapak.com/it/
4	CESAB CARRELLI ELEVATOR	IT04217660374	282202	315.609.000	434	http://www.cesab-forklifts.eu/En/pages/default.aspx
5	DRILLMEC S.P.A.	IT02316330402	289202	206.589.000	415	http://www.drillmec.com/
6	LOMBARDINI S.R.L.	IT01829970357	281111	180.789.000	584	http://www.lombardinigroup.it/it/homepage
7	SYSTEM S.P.A.	IT02438830362	282000	168.193.000	508	http://www.system-group.it/ita/
8	BREVINI POWER TRANSMISS	IT00262750359	281510	139.027.000	617	http://www.brevini.com/
9	MODULA S.P.A.	IT03197890365	282209	131.192.000	241	http://www.modula.eu/ita/
10	EMAK S.P.A.	IT00130010358	282400	128.602.000	393	http://www.emak.it/
11	SOILMEC - SOCIETA' PER AZI	IT00139200406	299209	123.963.000	420	http://www.soilmec.com/en/
12	CORGHI S.P.A.	IT01700320359	289993	121.250.000	684	http://www.corghi.com/default.php?&lang=IT
13	WALVOIL S.P.A.	IT01523540357	281400	120.732.000	766	http://www.walvoil.com/ita/azienda_mission.htm
14	ITALTRACTOR ITM S.P.A.	IT02929550362	283010	115.025.000	573	http://www.group-itm.com/Default.aspx
15	DIECI S.R.L.	IT01682740350	282209	107.212.000	204	http://www.dieci.com/lang_it/index.php
16	MARINI S.P.A.	IT00174890392	289000	103.501.000	377	http://www.marini.fayat.com/
17	DANFOSS POWER SOLUTION	IT07874070019	281200	93.774.000	265	http://www.danfoss.it/home/
18	CASAPPA S.P.A.	IT00717660344	281200	87.325.000	464	http://www.casappa.com/ita/home.htm
19	OCME SOCIETA' A RESPONS	IT00786410340	282930	87.161.000	424	http://www.ocme.it/website/default.aspx
20	A.M.A. S.P.A.	IT00639260355	283000	84.045.000	363	http://www.ama.it/

Figura 2.1: alcune delle aziende contenute nel file.

Si tratta di un file con una struttura piuttosto omogena. Ogni riga infatti contiene sempre lo stesso numero di informazioni.

Alcuni di questi dati non sono fondamentali: sicuramente sono d'interesse il nome dell'azienda e quello del relativo sito web, che serviranno per accedere alle pagine. Eventualmente si potrebbe tenere conto anche del fatturato e del numero di dipendenti, per fare una classificazione in base alle dimensioni dell'azienda.

Nel file sono presenti 100 aziende tutte con codice ateco 28. Verrà comunque predisposta una struttura in grado di contenere un numero arbitrio di aziende.

Per leggerlo sarà sufficiente creare per ciascuna azienda una entry in un database predisposto alla memorizzazione della stessa e delle relative informazioni.

2) File KPI

In **Figura 2.2** viene mostrato il file contenente i KPI.

Processo	KPI	Additional keywords				
CERTIFICAZIONI AMBIENTALI	ISO 14001					
	EMAS	Environmental Management and Audit Scheme				
	ISO 50001					
	ETICHETTA ECOLABEL					
	LCA	Life Cicle Assessment				
	FSC					
	GOLDPOWER					
	LEED					
CERTIFICAZIONI SOCIALI	SA 8000					
	ISO 26000					
	OHSAS 18001					
	IFS					
	ISO 22005					
	ISO 22000					
ENERGIA	UTILIZZO ENERGIE RINNOVABILI	pale eoliche	mulino ad acqua	mulino eolico	impianto co-generazione	pannelli solari
	POLITICHE DI RISPARMIO ENERGETICO	lampade a risparmio energetico	cappotto esterno	tetti verdi		
	RIDUZIONE DEI COSTI	Indicazione di minori costi e risparmi economici dovuti ad utilizzo di energie rinnovabili				
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI					
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	consumo energia	utilizzo energia			
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK					

Figura 2.2: alcuni dei processi e KPI contenuti nel file.

A differenza del file precedente, questo è sicuramente più complesso perché il numero di informazioni (e quindi di colonne) per ciascuna riga è variabile: infatti ad ogni KPI possono essere state aggiunte ulteriori parole chiave per facilitare e/o migliorare la ricerca.
Il file contiene 56 KPI raggruppati in 11 processi aziendali: sono quelli definiti al punto 1.3.2 tranne gli indicatori legati alle informazioni generali. Sarà necessario predisporre un database in grado di supportare questa suddivisione.
Per ogni riga letta si dovrà verificare se è contenuto un nuovo processo; se è presente si inserirà prima una nuova entry nel database, quindi si leggerà il KPI (con le relative parole addizionali) che verrà aggiunto al processo stesso. Se invece non c'è, verrà svolta solo la seconda parte legata al KPI.
Per quanto riguarda le parole chiave aggiuntive, non è possibile sapere a priori né quante sono né se sono presenti, perciò sarà sempre necessario creare una struttura adatta a contenerle. La lettura di ciascuna riga proseguirà finché verranno trovate ulteriori parole chiave, le quali verranno aggiunte al relativo KPI di appartenenza.

3) File Rilevazioni

In **Figura 2.3** viene mostrato il file contenente le rilevazioni, ma non di tutti i KPI per motivi di spazio. Il file infatti contiene una colonna per ciascun KPI.

	n° progressivo KPI	9	10	11	12	13	14	15	16	17	18	19
1	SACMI COOPERATIVA MECCANICI IMOLA SOCIETA' COOPERATIVA IN BREVE SACMI IMOLA S.C.	1	0	0	0	0	0	0	0	0	0	1
2	BONFIGLIOLI RIDUTTORI S.P.A.	1	0	0	0	0	0	0	0	0	0	1
3	MODULA S.P.A.	1	0	0	0	0	0	0	0	0	0	1
4	CORGHI S.P.A.	0	0	0	0	0	0	0	0	0	0	0
5	NORDMECCANICA S.P.A.	0	0	0	0	0	0	0	0	0	0	0
6	SOCIETA' PER AZIONI CURTI-COSTRUZIONI MECCANICHE	1	0	0	0	0	0	0	0	0	0	1
7	Terex Italia S.R.L.	0	0	0	0	0	0	0	0	0	0	0
8	DULEVO INTERNATIONAL S.P.A.	0	0	0	0	0	0	0	0	0	0	0
9	WERTHER INTERNATIONAL S.P.A.	0	0	0	0	0	0	0	0	0	0	0
10	Celli S.P.A.	0	0	0	0	0	0	0	0	0	0	0

Figura 2.3: alcune delle rilevazioni delle aziende.

Il file ha una struttura omogenea: ogni riga contiene il nome della azienda e un valore che indica se il KPI è effettivamente presente o no. L'ordine degli indicatori è lo stesso usato anche nel precedente file.
Il file contiene le rilevazioni di tutte 100 le aziende; ciascuna riga conterrà i valori di tutti 56 gli indicatori.
La lettura non sarà complicata: per associare le rilevazioni ad una azienda, sarà sufficiente cercare la riga con lo stesso nome e quindi leggere tutti i valori, associandoli al rispettivo KPI.

2.2.2 Download e lettura dei siti web

La seconda fase tratta lo scaricamento dei siti. Nonostante sia possibile accedere direttamente alle pagine web a run time, è molto più conveniente salvarle localmente in precedenza.
Questo garantirà una flessibilità e soprattutto un'efficienza migliore perché il download dei siti web verrà effettuato una sola volta. Infatti il download potrebbe richiedere molto tempo, in quanto esso aumenta proporzionalmente col numero di pagine da scaricare e potrebbe anche interrompersi in caso la connessione ad Internet dovesse saltare. Non si avranno quindi vincoli legati all'esecuzione della ricerca: si potrà rieseguire più volte senza l'attesa di ulteriori

download. Inoltre l'accesso da filesystem è più veloce rispetto che leggere i dati direttamente dalla pagina web.
La seconda fase di divide in due parti:

- Download dei siti
- Lettura delle pagine scaricate

Nella prima parte viene realmente effettuato il download. Per ciascun sito scaricato verrà creata una cartella contenente tutte le relative pagine.
Nella seconda parte invece effettuata la lettura delle pagine: verrà estratto il contenuto di ogni pagina, il quale sarà memorizzato nel software all'azienda di appartenenza.
L'obiettivo sarà quello di rendere il tutto automatico: per ciascuna azienda, leggendo il nome del sito web, il software scaricherà tutte le pagine che lo compongono e, dopo averne estratto il contenuto, assocerà quest'ultimo all'azienda corretta.
Dal momento che le pagine web utilizzano il formato **HTML**, sarà necessario un parser specifico per leggere questo tipo di file.
Acronimo di HyperText Markup Language, in italiano linguaggio a marcatori per ipertesti, HTML è il linguaggio di markup usato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web sotto forma di pagine web.
Un ipertesto è un insieme di documenti messi in relazione tra loro per mezzo di parole chiave. Può essere visto come una rete; i documenti ne costituiscono i nodi. La caratteristica principale di un ipertesto è che la lettura può svolgersi in maniera non lineare: qualsiasi documento della rete può essere "il successivo", in base alla scelta del lettore di quale parola chiave usare come collegamento.
Il linguaggio HTML ha come scopo quello di gestire i contenuti associandone o specificandone allo stesso tempo la struttura grafica (layout) all'interno della pagina web da realizzare grazie all'utilizzo di tag. Ogni tag specifica un diverso ruolo dei contenuti che esso contrassegna. La formattazione consiste nell'inserimento nel testo di marcatori o etichette, detti tag, che descrivono caratteristiche come la funzione, il colore, le dimensioni, la posizione relativa all'interno della pagina.
Il componente principale della sintassi di questo linguaggio è l'elemento, inteso come struttura di base a cui è delegata la funzione di formattare i dati o indicare al browser delle informazioni.

Ogni elemento è racchiuso all'interno di marcature dette tag, costituite da una sequenza di caratteri racchiusa tra due parentesi angolari o uncinate (< >), cioè i segni minore e maggiore (Es.:
, che serve ad indicare un ritorno a capo).
Il primo elemento di un documento HTML è la definizione del tipo di documento (Document Type Definition o DTD): serve al browser per identificare le regole di interpretazione e visualizzazione da applicare al documento.
Dopo la dichiarazione del DTD, il documento HTML presenta una struttura ad albero annidato, composta da sezioni delimitate da tag opportuni che al loro interno contengono a loro volta sottosezioni più piccole, sempre delimitate da tag.
La struttura più esterna è quella che delimita l'intero documento, eccetto la DTD, ed è compresa tra i tag <html> e </html>. All'interno di questi tag, lo standard prevede sempre la definizione di due sezioni ben distinte e disposte in sequenza ordinata:

- La sezione di intestazione o header, delimitata tra i tag <head> e </head>, che contiene informazioni di controllo normalmente non visualizzate dal browser, con l'eccezione di alcuni elementi
- La sezione del corpo o body, delimitata tra i tag <body> e </body>, che contiene la parte informativa vera e propria, ossia il testo, le immagini e i collegamenti che costituiscono la parte visualizzata dal browser.

Al di sotto di questa suddivisione generale, lo standard non prevede particolari obblighi per quanto riguarda l'ordine e/o il posizionamento delle ulteriori sottosezioni all'interno sia dell'header che del body. La **Figura 2.4** mostra la suddivisione appena descritta.

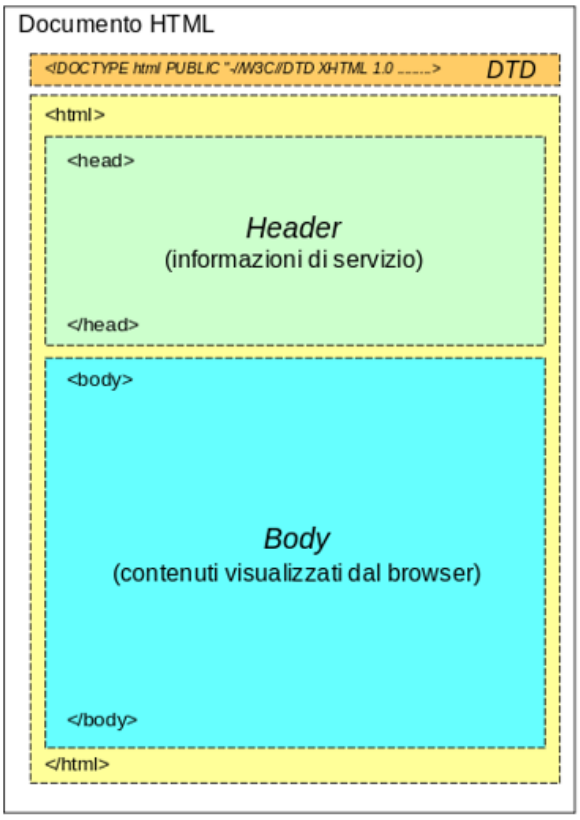

Figura 2.4: struttura di un documento HTML.

Infine una pagina HTML può essere rappresentata come una struttura ad albero che prende il nome di DOM, ovvero Document Object Model. Quando un browser carica una pagina HTML la scompone e costruisce la struttura ad albero del DOM. La **Figura 2.5** mostra chiaramente questa differenza.

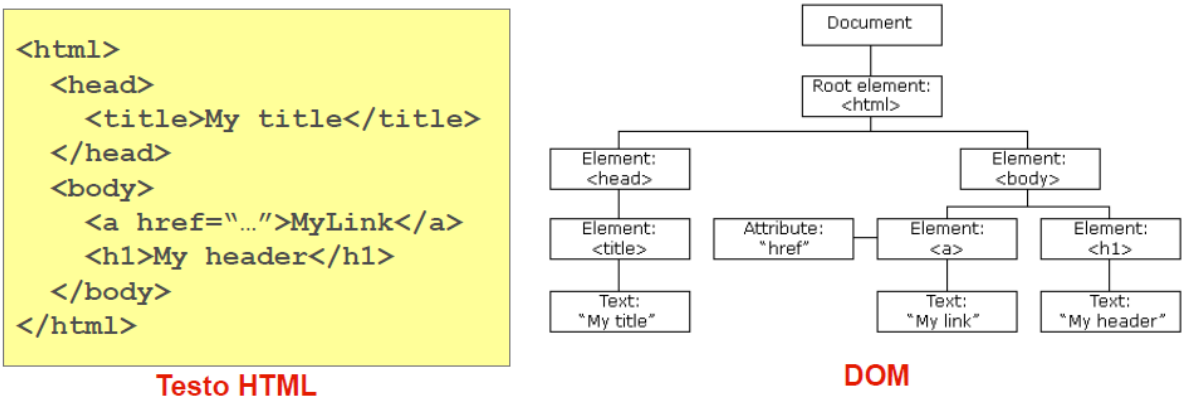

Figura 2.5: differenza tra il codice HTML e il DOM di un documento.

2.2.3 Processing dei dati

Il terzo aspetto legato all'elaborazione dei dati ed è sicuramente quello centrale del problema. Verrà infatti discusso un possibile approccio all'algoritmo di ricerca.

L'idea di base per la ricerca dei KPI è di estrarre dalle pagine web tutto il contenuto testuale e controllare se compaiono.

In seguito all'analisi della struttura di una pagina HTML svolta nella precedente fase, è intuibile che l'algoritmo si concentrerà principalmente sul body delle pagine: qui infatti è contenuto tutto il testo che viene realmente mostrato nella pagina web e perciò è la parte più significativa in cui cercare i KPI.

A primo impatto si potrebbe semplicemente controllare se all'interno del testo compaiono i KPI. Dall'analisi degli input si nota che gli indicatori possono differire molto tra di loro: alcuni sono composti da poche parole chiave, mentre altri sono delle vere e proprie frasi di senso compiuto.

Per quanto riguarda i primi non si dovrebbero riscontrare grossi problemi, ma per i secondi la situazione è più complessa e richiede un'analisi più attenta: infatti sarà molto difficile ritrovare la stessa identica frase all'interno di una pagina. È necessario quindi un algoritmo che vada oltre l'idea inizialmente pensata.

Per risolvere questo problema si deve "spezzare" ogni KPI nelle singole parole che lo compongono. Quindi l'algoritmo conterà quante delle parole che compongono l'indicatore compaiono e lo memorizzerà assieme al numero totale delle stesse.

Difficilmente però verranno trovate tutte le parole, soprattutto per i KPI più lunghi e discorsivi, in quanto potrebbero essere usati sinonimi, tempi verbali diversi, ecc. Si dovrà definire una soglia minima di accettazione, cioè definire un numero di parole che devono essere trovate affinché si possa considerare come presente un indicatore.

Non verrà fatta distinzione tra il KPI e le eventuali parole addizionali: verranno infatti trattati allo stesso modo. Questo significa che se una delle parole aggiuntive dovesse essere trovata dall'algoritmo, il relativo KPI verrà segnato come presente.

2.2.4 Visualizzazione e confronto dei risultati

L'ultimo passo riguarda la visualizzazione dei risultati e il confronto con le misurazioni reali. Il software dovrà mostrare i risultati di ciascuna azienda, indicando quali KPI sono stati trovati e quali no e il confronto con le rilevazioni reali, indicando la percentuale di correttezza sia dei singoli processi che di tutti gli indicatori nel totale.

Successivamente verranno mostrati alcuni dati riassuntivi di tutte le aziende processate, specificando la media generale di precisione. Inoltre verrà fatta una classificazione in base alle dimensioni delle aziende, in modo da mostrare se questo indice o meno sulla correttezza dei risultati.

Infine saranno visualizzate alcune informazioni legate alle performance del software: ovvero i tempi di processamento e l'occupazione in memoria di ciascuna delle quattro fasi, sia per ogni singola azienda che una media calcolata su tutte.

Capitolo 3 Strumenti utilizzati

Per quanto riguarda gli strumenti esterni che verranno usati, oltre al linguaggio Java per la programmazione, viene fatta distinzione tra software e librerie.

3.1 Software

Un software è l'informazione o le informazioni utilizzate da uno o più sistemi informatici e memorizzate su uno o più supporti informatici. Tali informazioni possono essere quindi rappresentate da uno o più programmi, da uno o più dati, oppure da una combinazione delle due. Software è un termine generico che definisce programmi e procedure utilizzati per far eseguire ad un computer un determinato compito. Generalmente viene fatta distinzione tra:

- Software di base o di sistema, in quanto indispensabile al funzionamento del computer dal momento che senza di esso non sarebbe che hardware inutilizzabile. Un esempio è il sistema operativo.
- Software applicativo, che comprende i programmi che lavorano sfruttando le prestazioni che offre il sistema operativo. Fanno parte di questa categoria le applicazioni gestionali destinati alle esigenze specifiche di un utente o di un'azienda.

3.1.1 GNU Wget

Wget è un software per il recupero di file utilizzando i protocolli Internet più diffusi, quali HTTP, HTTPS e FTP. È uno strumento a riga di comando non interattivo, quindi può essere facilmente richiamato da script o da chiamate pianificate tramite crontab.
Si tratta di uno strumento davvero molto potente, in grado di effettuare il recupero di file di grandi dimensioni o il mirroring di interi siti Web o FTP.

Alcune delle caratteristiche che è in grado di offrire sono:

- Riprendere i download interrotti, utilizzando REST e GAMMA
- Usare nomi di file wild card e ricorsivamente le directory specchio
- Convertire i collegamenti assoluti nei documenti scaricati sul relativo, in modo che i documenti scaricati possono collegare gli uni agli altri a livello locale
- Compatibilità con la maggior parte dei sistemi operativi UNIX e Microsoft Windows
- Supporto ai proxy HTTP
- Supporto ai cookie HTTP
- Supporto alle connessioni HTTP persistenti
- Utilizzare il timestamp dei file locali per determinare se i documenti devono essere ri-scaricati.

A default, Wget è molto semplice da invocare. La sintassi basi è:

`wget [option]… [URL]…`

Wget procederà semplicemente a scaricare tutti gli URL specificati nella riga di comando. URL, acronimo di Uniform Resource Locator, è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet, come ad esempio un documento, un'immagine, un video, rendendola accessibile ad un client che ne faccia richiesta attraverso l'utilizzo di un web browser.
In parole povere un URL è quello che comunemente viene chiamato “nome del sito web” e che viene digitato nella barra del browser per accedere ad un determinato sito.
È possibile cambiare alcuni dei parametri impostati di default. Si può fare in due modi: permanentemente, aggiungendo i comandi al file “.wgetrc”, oppure specificando le modifiche direttamente durante l’invocazione.
Si possono modificare molti parametri del download, tramite una serie di opzioni suddivise in varie categorie, tra le quali:

- Formato dell’URL
- Opzioni sintattiche

- Opzioni di avvio di base
- Logging e opzioni dei file di input
- Opzioni di download
- Opzioni delle directory
- Opzioni HTTP, HTTPS e FTP
- Opzioni per il download ricorsivo

Wget verrà usato nella fase di download. Trattandosi di un software di tipo applicativo a riga di comando, sarà facile integrarlo all'interno del codice del software.

3.2 Librerie

Una libreria è un insieme di funzioni o strutture dati predefinite e predisposte per essere collegate ad un programma software attraverso un opportuno collegamento.
Lo scopo delle librerie software è di fornire una collezione di entità di base pronte per l'uso, evitando al programmatore di dover riscrivere ogni volta le stesse funzioni o strutture dati e facilitando così le operazioni di sviluppo e manutenzione. Questa caratteristica si inserisce quindi nel più vasto contesto del "richiamo di codice" all'interno di programmi e applicazioni ed è presente in quasi tutti i linguaggi. I vantaggi principali derivanti dall'uso di un simile approccio sono i seguenti:

- Si può separare la logica di programmazione di una certa applicazione da quella necessaria per la risoluzione di problemi specifici, quali il calcolo di funzioni matematiche o la gestione di collezioni.
- Le entità definite in una certa libreria possono essere riutilizzate da più applicazioni.
- Si può modificare la libreria separatamente dal programma, senza limiti alla potenziale vastità di funzioni e strutture dati man mano disponibili nel tempo.

3.2.1 Jsoup

Jsoup è una libreria Java che consente di lavorare con i file HTML. Fornisce una serie di API per estrarre e manipolare i dati, sfruttando al meglio DOM, CSS e script JQuery.
Implementa le specifiche WHATWG HTML5 e parsifica le pagine HTML sfruttando il DOM come i browser moderni. Inoltre è in grado di trattare tutte le varietà di HTML esistenti.
Alcune delle funzionalità offerte sono:

- Parsifica HTML da URL, file o stringa.
- Ricerca ed estrazione dei dati sfruttando sia il DOM che i selettori CSS.
- Manipolazione di elementi, attributi e testo HTML.
- Pulizia dell'input tramite filtri basati su White-list per prevenire attacchi XSS.

Il metodo base di questa libreria è "parse(String page)", dove *page* è una stringa (ovvero una sequenza di caratteri ben formata) contenente codice in linguaggio HTML.
È possibile leggere dati anche da file, tramite il metodo "parse(File in, String charsetName)": "in" deve essere il nome di un file esistente nel filesystem, mentre "charsetName" è il set di caratteri usato nel file.
Infine è possibile leggere una pagina HTML direttamente dal web, senza alcun download precedente, tramite il metodo "connect(String url).get()": il parametro "url" è ovviamente l'URL del sito web che si vuole leggere, mentre get() consente di effettivamente l'accesso alla pagina.
Il risultato dell'invocazione di tutte le funzioni appena descritte (parse e connect) è sempre un oggetto di tipo Document. In esso viene salvato il DOM della pagina letta assieme a molte altre informazioni. Tramite l'oggetto Document si può quindi ripercorre l'albero sintattico della pagina letta.
Il metodo "body()" applicato ad un Document consente di estrarre il contenuto del body, che viene restituito come stringa.
Jsoup verrà usata nella fase di download, in particolare nella seconda parte, per la lettura dei file HTML.

3.2.2 OpenCsv

OpenCsv è una libreria Java che offre un parser per file di tipo CSV. Supporta tutte le operazioni fondamentali che generalmente si fanno con i file CSV:

- Numero arbitrario di valori per riga.
- Ignora le virgole all'interno di elementi virgolettati.
- Gestisce gli elementi tra virgolette anche se contengono carriage return.
- Separatori e caratteri di citazione sono configurabili.
- Legge tutte le righe con un comando, oppure si può agire sulle singole.
- Crea file CSV da una lista di stringhe ben formate.

Per leggere un file CSV con questa libreria è necessario innanzitutto creare un nuovo oggetto CSVReader: per farlo è necessario passare almeno un parametro di tipo FileReader, che a sua volta richiede il nome del file che si vuole aprire.
Per estrarre il contenuto è sufficiente eseguire il metodo "readAll()", invocato sull'oggetto di tipo CSVReader appena creato, che restituisce una lista di stringhe dove ciascuna rappresenta una delle righe del file letto.

```
CSVReader reader = new CSVReader(new FileReader("yourfile.csv"));
List myEntries = reader.readAll();
```

È possibile anche leggere una singola riga tramite il metodo "readNext()", sempre invocato su un oggetto CSVReader.

```
CSVReader reader = new CSVReader(new FileReader("yourfile.csv"), '\t');
```

La libreria supporta anche l'utilizzo di separatori non standard (ovvero diversi da , e ;): per fare ciò è necessario, quando si crea l'oggetto CSVReader passare un secondo parametro contenente il separatore usato nel file.
OpenCsv verrà usata nella prima fase di gestione degli input per parsificare tutti e tre i file.

Capitolo 4 Implementazione e confronto

Si tratterà qui l'implementazione del software, cercando di seguire le linee guida definite durante l'analisi discussa nel Capitolo 2. Al termine verrà mostrato un confronto con i dati rilevati a mano e si analizzeranno eventuali criticità.

4.1 Progettazione e realizzazione

L'implementazione viene suddivisa nelle 4 macro fasi precedentemente introdotte durante l'analisi. In **Figura 4.1** viene mostrato il diagramma contenente le principali funzionalità di ciascuna fase.

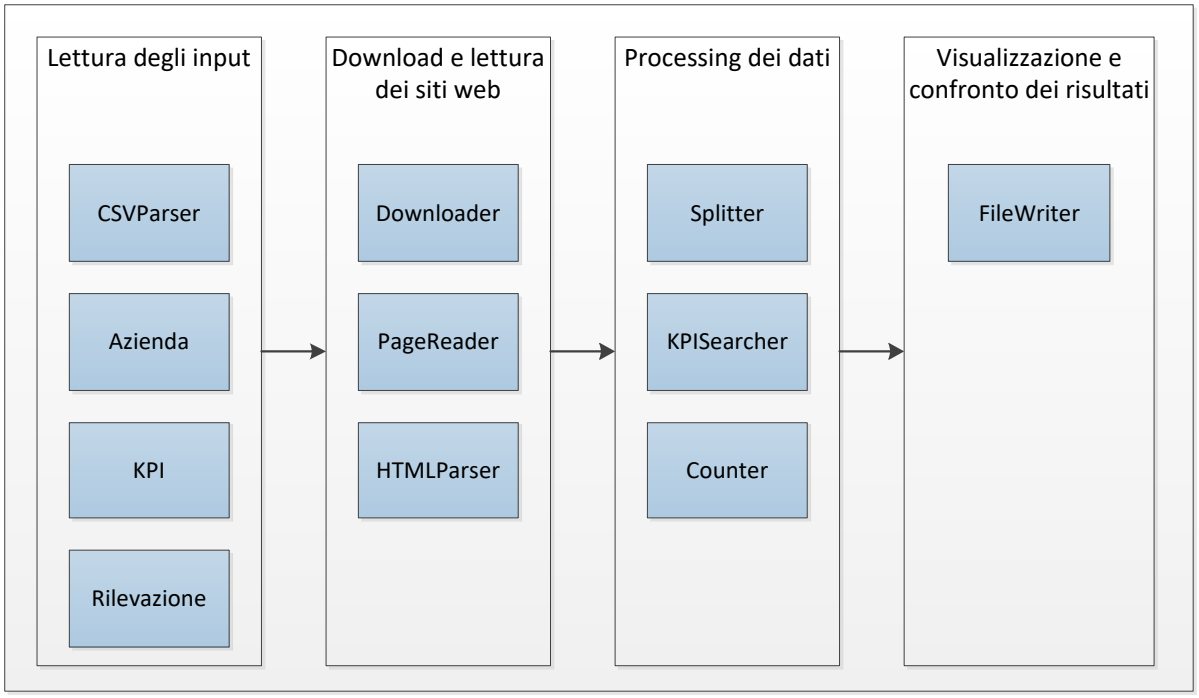

Figura 4.1: diagramma delle principali funzionalità del software.

All'avvio il software legge il file contenente le aziende e associa a ciascuna i KPI e le rispettive rilevazioni, leggendo gli altri due file. Quindi processa una ad una le aziende e per ciascuna:

- Controlla se il sito è già stato scaricato, altrimenti procede con il download. Poi legge le pagine e le memorizza.
- Ricerca i KPI all'interno delle pagine scaricate, cercando le singole parole che compongono l'indicatore e contando le occorrenze.
- Visualizza i risultati e li confronta con i valori trovati a mano.

Eccetto la prima fase che viene eseguita una volta sola all'avvio, l'esecuzione delle altre è sequenziale: viene processata un'azienda per volta, svolgendo tutte le tre fasi per ciascuna prima di passare alla successiva.

4.1.1 Lettura degli input

Innanzitutto si procede con la conversione di tutti e tre i file in formato CSV. Questa operazione può essere fatta tramite Excel stesso, eseguendo l'operazione "salva con nome" e selezionando come tipo di file ".csv" anziché ".xls" o ".xlsx". In tal modo diventano facilmente leggibili tramite la libreria OpenCsv. In **Figura 4.2** viene mostrato il codice per la lettura dei tre file.

```
//Legge le aziende dal file csv
List<Azienda> aziende = CsvParser.leggiAziende("Siti.csv");

//Per ogni azienda leggi i due file
for (Azienda a : aziende) {
    CsvParser.leggiKPI("KPI.csv", a);
    CsvParser.leggiRilevazioni("Aziende.csv", a);
}
```

Figura 4.2: codice per la lettura dei tre file.

Viene prima letto il file contenente le aziende, quindi a ciascuna di esse viene associata la lista di KPI e le rilevazioni.

1) Lettura delle aziende

Per prima cosa viene definita la classe Azienda composta da nome, partita iva e nome del sito web, tutti e tre stringhe, più il numero dei dipendenti come intero. Inoltre contiene anche la lista dei processi (con i relativi KPI) e la lista delle pagine del sito web.
Il software esegue quindi il metodo "leggiAziende(String fileName)", che effettua la lettura delle aziende presenti all'interno del file passato tramite il parametro "fileName" e restituisce una lista di oggetti Azienda. Viene mostrato in **Figura 4.3**.

```
public static List<Azienda> leggiAziende(String fileName) {
    List<Azienda> aziende = new LinkedList<Azienda>();
    try {
        //Parsing del file
        CSVReader reader = new CSVReader(new FileReader(fileName), ';');

        //Scarta le prime righe che non contengono aziende
        reader.readNext();
        reader.readNext();
        reader.readNext();
        reader.readNext();
        reader.readNext();

        String[] nextLine;
        Azienda azienda = null;
        while ((nextLine = reader.readNext()) != null) {
            azienda = new Azienda(nextLine[1], nextLine[2], nextLine[6]);
            aziende.add(azienda);
        }
        //Chiude il reader
        reader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return aziende;
}
```

Figura 4.3: codice del metodo "leggiAziende()".

Il metodo crea innanzitutto una lista di oggetti Azienda, per ospitare i dati che verranno letti. La lista è implementata come LinkedList: detta anche "lista concatenata", si tratta di un elenco "collegato", ovvero ogni elemento che viene aggiunto avrà un indice crescente che parte da 0 e due riferimenti che puntano uno all'elemento successivo e l'altro al precedente.

Si procede con la creazione di un oggetto CSVReader, passando al costruttore i parametri il nome del file da leggere e il separatore usato all'interno dello stesso, in questo caso ";" ovvero lo standard utilizzato da Excel.
Con la funzione "readNext()" vengono scartate le prime 5 righe, che non contengono nessun dato da leggere. Quindi viene eseguito un ciclo sulle restanti: ad ogni passaggio si legge la riga successiva e, se questa esiste, viene creato un nuovo oggetto Azienda utilizzando 4 parametri che rappresentano rispettivamente il nome, la partita iva, i dipendenti e il nome del sito web (che corrispondono alla colonna 2, 3, 5 e 7 del file). Al termine viene aggiunto l'oggetto appena creato alla lista delle aziende. Infine, terminata la lettura, si chiude il CSVReader per liberare spazio e viene restituita la lista di aziende.

2) Lettura dei KPI

Per la lettura di questo file è necessario definire due classi:

- KPI, composta da un nome e una lista di parole addizionali
- Processo, composta da un nome e una lista di KPI

Il metodo "leggiKPI(String fileName, Azienda a)" effettua la lettura dei KPI e dei processi presenti all'interno del file passato come primo parametro e li associa all'azienda passata come secondo. Viene mostrato in **Figura 4.4**.

```java
public static void leggiKPI(String fileName, Azienda a) {
    //Mappa ogni processo con una lista di KPI
    //Ogni KPI a sua volta contiene un nome e una lista di additional keywords
    try {
        //Parsing del file
        CSVReader reader = new CSVReader(new FileReader(fileName), ';');
        //Scarta la prima riga che contiene i nomi delle colonne
        reader.readNext();

        String[] nextLine;
        Processo processo = null;
        while ((nextLine = reader.readNext()) != null) {
            //Se trova un nuovo processo, crea una un nuovo oggetto
            if (!nextLine[0].equals("")) {
                processo = new Processo(nextLine[0]);
                a.getProcessi().add(processo);
            }
            //Legge il kpi
            KPI newKpi = new KPI(nextLine[1]);
            //Aggiunge le additional keywords al kpi
            for (int i=2; i<nextLine.length; i++) {
                if (!nextLine[i].equals("")) {
                    newKpi.aggiungiKeyword(nextLine[i]);
                }
            }
            //Inserisce il kpi nella lista del processo
            processo.getListaKPI().add(newKpi);
        }
        //Chiude il reader
        reader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 4.4: codice del metodo "leggiKPI()".

Innanzitutto il metodo crea un oggetto CSVReader e scarta la prima riga che non contiene informazioni.

Procede leggendo riga per riga: se il primo campo contiene del testo, significa che è presente un nuovo processo e perciò istanzia un nuovo oggetto Processo, che viene aggiunto alla lista dei processi dell'azienda. Quindi crea sempre un oggetto KPI usando come parametro la seconda colonna del file, che contiene il nome. Poi esegue un ciclo sui restanti campi (nextLine.length è pare al numero delle colonne): se non sono vuoti, significa che ci sono parole addizionali e le aggiunge all'oggetto KPI appena creato. Infine aggiunge il KPI al processo e chiude il CSVReader.

3) Lettura delle rilevazioni

In questo caso non servono nuove classi: le informazioni lette verranno aggiunte agli oggetti KPI di ciascuna azienda.

In **Figura 4.5** è mostrato il metodo "leggiRilevazioni(String fileName, Azienda a)", il quale effettua la lettura delle rilevazioni del file passato come primo parametro, estraendo quelle relative all'azienda passata come secondo parametro. Quindi le associa ai rispettivi KPI dell'azienda.

La lettura viene semplificata utilizzando come ordine dei valori lo stesso usato per i KPI: grazie all'utilizzo delle LinkedList, non è necessario il controllo del nome ma è sufficiente l'indice. La terza colonna corrisponde sempre al primo KPI, la quarta al secondo, e così via fino all'ultimo.

```
public static void leggiRilevazioni(String fileName, Azienda a) {
    try {
        //Parsing del file
        CSVReader reader = new CSVReader(new FileReader(fileName), ';');

        String[] nextLine;
        while ((nextLine = reader.readNext()) != null) {
            //Cerca la riga con lo stesso nome dell'azienda
            if (a.getNome().toLowerCase().contains(nextLine[1].toLowerCase())) {
                int count = 2;
                //Associa a ciascun KPI la rilevazione
                for (Processo p : a.getProcessi()) {
                    for (KPI k : p.getListaKPI()) {
                        k.setRilevazone(Integer.parseInt(nextLine[count]));
                        count++;
                    }
                }
                break;
            }
        }
        // Chiudo il reader
        reader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 4.5: codice del metodo "leggiRilevazioni()".

Innanzitutto il metodo crea un oggetto CSVReader e scarta la prima riga che non contiene informazioni.

Procede leggendo riga per riga finché non trova quella con nome (seconda colonna) uguale a quello dell'azienda: quindi legge tutti i valori e li associa al rispettivo KPI. Tramite break esce subito dal ciclo in quanto ha già letto le rilevazioni e ulteriori passaggi sono inutili. Infine chiude il CSVReader.

4.1.2 Download e parsing dei siti web

Una volta terminata la lettura dei file si procede con il download. Tutti i siti vengono scaricati in una directory principale, ciascuno in una cartella dedicata contenente tutte le rispettive pagine. In **Figura 4.6** viene mostrato il codice.

```
String mainFolderName = "Source";
File mainFolder = new File(mainFolderName);
File[] siteList = mainFolder.listFiles();
if (sitoDaScaricare(a.getSito(), siteList)) {
    Downloader.scaricaSito(a.getSito());
}
```

Figura 4.6: codice per il download dei siti.

Il parametro "mainFolderName" rappresenta la cartella principale contenente tutti i siti scaricati.

```
private static boolean sitoDaScaricare(String sito, File[] siteList) {
    for (File f : siteList) {
        if (f.isDirectory() && sito.contains(f.getName())) {
            return false;
        }
    }
    return true;
}
```

Figura 4.7: codice del metodo "sitoDaScaricare()".

Il metodo "sitoDaScaricare(String sito, File[] siteList)" (**Figura 4.7**) controlla che il sito dell'azienda selezionata non sia già stato scaricato, verificando se esiste una cartella con il nome del sito, altrimenti viene effettuato il download tramite il metodo "scaricaSito(Azienda a)" (**Figura 4.8**).

```
public static void scaricaSito(String nomeSito) {
    //Definisce il comando di download
    String command = "Source/wget -E -x -k -P Source "
            + "-A html,htm,php,asp,aspx -r -l 50 ";
    //Ottiene un oggetto Runtime
    Runtime rt = Runtime.getRuntime();
    try {
        //Esegue il download del sito "nomeSito"
        Process proc = rt.exec(command + nomeSito);
        //Finché il download è attivo attende
        while (proc.isAlive());
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Figura 4.8: codice del metodo "scaricaSito()".

Tramite "getRuntime()" si ottiene un oggetto di tipo Runtime, che serve ad eseguire il comando di download definito nella stringa "command".
Il metodo "exec(String name)" consente di eseguire il comando passato come parametro, in questo caso composto dalle stringhe "command" più "nomeSito" (ovvero il nome del sito web da scaricare).
La sintassi di "command" è:

```
wget -E -x -k -P Source -A html,htm,asp,aspx,php -r -l 50 [URL]
```

dove al posto di [URL] viene messo il sito del quale si vogliono scaricare le pagine (tramite la stringa nomeSito).
Per quanto riguarda le singole opzioni:

- -E, serve per forzare l'aggiunta dell'estensione ".html" a quei file di tipo "application/xhtml+xml" o "text/html" il cui URL non termina con la regular expression "`\.[Hh][Tt][Mm][Ll]?`". Questo serve per i siti che usano le pagine in asp/aspx, oppure quando si tratta di materiale generato tramite CGI.
- -x, serve per forzare la creazione di una gerarchia di directory, anche nel caso in cui non sia prevista.
- -k, consente di convertire i link assoluti all'interno dei documenti in relativi, in modo tale da creare una struttura navigabile totalmente in locale.

- -A, definisce una serie di estensioni che devono essere scaricate; tutte i file che terminano con altri suffissi non verranno scaricati.
- -P, consente di indicare dove salvare il download.
- -r, rende il download ricorsivo, consentendo di scaricare tutte le pagine collegate all'URL indicato.
- -l, consente di impostare il livello massimo della ricorsione.

Quindi "command" seguito da "sito" consente di eseguire il download secondo le modalità appena descritte.
Una volta terminato il download, si procede con l'inserimento dei dati scaricati nel programma. Il codice è mostrato in **Figura 4.9**.

```
// Cerca la directory che contiene le pagine
for (File f : siteList) {
    if (f.isDirectory() && a.getSito().contains(f.getName())) {
        // Lettura ricorsiva delle pagine data la directory principale
        PageReader.leggiPagine(a, f.getAbsolutePath());
    }
}
```

Figura 4.9: codice del metodo per la lettura delle pagine.

Si esegue un ciclo su tutti i file presenti nella directory principale e su di essi viene fatto un doppio controllo; ogni file:

- Deve essere una directory.
- Il nome deve coincidere con quello del sito.

Se il controllo da esito positivo, viene invocata la funzione ricorsiva "leggiPagine(Aziende a, String fileName)" (**Figura 4.10**), che prende come primo parametro l'azienda corrente e come secondo il percorso assoluto della cartella contenente i file.

```java
public static void leggiPagine(Azienda a, String fileName) {
    File file = new File(fileName);
    //Se è una directory, esegue leggiPagine per tutti i file che trova
    if (file.isDirectory()) {
        File[] fileList = file.listFiles();
        for (File f : fileList) {
            leggiPagine(a, f.getAbsolutePath());
        }
    }
    else {
        //Se è un file HTML lo legge e lo aggiunge all'elenco di pagine
        if (file.isFile() && file.getName().endsWith(".html")) {
            a.getPagine().add(HtmlParser.LeggiFile(file));
        }
    }
}
```

Figura 4.10: codice del metodo "leggiPagine()".

Il metodo crea un oggetto File con il nome passato e controlla che sia una directory:

- Se è vero vengono letti tutti i file contenuti nella cartella e per ciascuno viene invocata nuovamente la "leggiPagine()".
- Se è falso controlla che sia un file e che l'estensione sia ".html", e se entrambe sono vere invoca la funzione "leggiFile(File input)".

```java
public static String LeggiFile(File input) {
    String output = "";
    try {
        //Parsifica il file creando il Document
        Document doc = Jsoup.parse(input, "UTF-8");
        //Estrae il contenuto testuale del body
        output = doc.body().text().toLowerCase();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return output;
}
```

Figura 4.11: codice del metodo "LeggiFile()".

Il metodo "LeggiFile(File input)" (**Figura 4.11**) procede leggendo il contenuto del file HTML passato e aggiungendo la pagina all'elenco memorizzato all'interno dell'oggetto Azienda.

Innanzitutto definisce una variabile “output” dove salvare il contenuto. Quindi crea un oggetto Document che ospiterà il DOM del file, il quale si ottiene tramite la funzione “parse(File input, String charset)”: essa prende come parametro il file e il **charset** usato nella pagina.
Una codifica di caratteri, o charset, consiste in un codice che associa un insieme di caratteri ad un insieme di altri oggetti, come numeri o pulsazioni elettriche, con lo scopo di facilitare la memorizzazione di un testo in un computer o la sua trasmissione attraverso una rete di telecomunicazioni. Esempi comuni sono la codifica ASCII e la Unicode.
In questo caso viene usato UTF-8, cioè codifica Unicode a 8 bit. Infine estrare il contenuto testuale del body, scartando quindi l’head, tramite le funzioni “body()” e “text()”. Inoltre il testo viene portato minuscolo tramite “toLowerCase()”.

4.1.3 Ricerca dei KPI

La terza parte è quella più importante, dove viene implementato l’algoritmo. Terminata la lettura delle pagine, il programma procede con il conteggio delle occorrenze. Per ciascuna azienda l’algoritmo cercherà i KPI all’interno delle pagine del relativo sito web. Il codice è mostrato in **Figura 4.12**.

```
for (Processo p : a.getProcessi()) {
    for (KPI kpi : p.getListaKPI()) {
        kpi.contaOccorrenze(a.getPagine());
    }
}
```

Figura 4.12: codice per il conteggio delle occorrenze.

Per ogni KPI contenuto all’interno dei processi vengono contate le occorrenze tramite il metodo “contaOccorrenze(List<String> pages)”, che riceve come parametro la lista delle pagine del sito web precedentemente inserite nel software. Viene mostrato in **Figura 4.13**.

```
public void contaOccorrenze(List<String> pages) {
    for (String page : pages) {
        conta(page);
    }
}
```

Figura 4.13: codice del metodo “contaOccorrenze()”.

Il metodo “contaOccorrenze()” non fa altro che chiamare per ogni pagina la funzione “conta(String page)”, che contiene l’algoritmo vero e proprio di ricerca. Quest’ultimo viene mostrato in **Figura 4.14**.

```
private void conta(String page) {
    for (String key : keywords) {
        String regexp = "[^\\p{L}\\p{Nd}]+";
        //Separa la key nelle singole parole, usando la regexp
        List<String> temp = Arrays.asList(key.toLowerCase().split(regexp));
        double count = 0;   //Contatore delle parole trovate
        double words = temp.size(); //Memorizza il totale delle parole
        for (int i=0; i<temp.size(); i++) {
            if (page.contains(temp.get(i))) {
                //Aumenta il contatore quando trova una parola
                count++;
            }
        }
        int def = temp.size() / 4;  //Approssimazione al 75%
        if (words >= 4)             //Ogni 4 parole ne "toglie" 1
            words = words - def;    //Se key ha 8 parole ne bastano 6
        if (count >= words) {
            occorrenze++;
            //Calcola quante parole sono state trovate e lo memorizza
            double precisione = count / temp.size();
            precisioni.put(key, new Double(precisione));
            totale += precisione;
        }
    }
}
```

Figura 4.14: codice del metodo “conta()”.

Il metodo “conta” esegue il conteggio del KPI e delle parole addizionali all’interno della pagina passata: come definito nell’analisi, tratta tutto allo stesso modo, sia i KPI veri e propri che le parole chiave aggiuntive: infatti sono tutte memorizzate in una lista di tipo String chiamata “keywords”.

Procede separando la parola/frase che deve essere cercata: per farlo sfrutta il metodo “split(String pattern)” di String, che come si può intuire dal nome consente di dividere una stringa tramite un “pattern” passato come parametro. Per fare ciò è necessaria una regular expression.

Una espressione regolare, in inglese regular expression e spesso abbreviata in regexp o RE, è una sequenza di simboli (quindi una stringa) che identifica un insieme di stringhe. Definisce una funzione che prende in ingresso una stringa, e restituisce in uscita un valore 1 o 0, a seconda che la stringa segua o meno un certo pattern.

In questo caso viene usata la regexp "`[^\\p{L}\\p{Nd}]+`": in parole povere consente di separare le parole di una frase qualunque sia il separatore (in pratica separa tutti i non caratteri e numeri). Ad esempio se in input viene data la stringa "Sono, 5; prova" si ottengono 3 stringhe, "Sono" "5" e "prova".

Prima spezzare il KPI, il metodo lo porta tutto in minuscolo tramite "toLowerCase()", come in precedenza è stato fatto anche per la pagina, in modo tale da trattare tutti i contenuti testuali con caratteri uguali. Se ad esempio si cerca l'indicatore "Codice Etico Aziendale", esso verrà trovato anche se scritto con maiuscole/minuscole diverse.

Una volta fatto ciò crea un campo "count" e lo inizializza a 0, che rappresenta il contatore di quante parole di quelle che compongono il KPI sono state trovate, e un campo "words" che rappresenta il numero totale di queste parole, cioè la dimensione della lista creata con lo split. Quindi cerca tramite un ciclo quali e quante di queste parole sono contenute nella pagina, incrementando il contatore in caso di match positivo per ciascuna parola trovata.

Prima di continuare è necessario chiarire un aspetto: come detto nell'analisi, difficilmente si troveranno tutte le parole che compongono un KPI, soprattutto nel caso in cui si tratti di una frase composta da molte parole. È necessario impostare una soglia, cioè definire quante parole sono sufficienti affinché il software possa "confermare" la presenza del KPI.

Verranno provate due percentuali: 50% e 75%. Nel primo caso devono comparire almeno la metà delle parole, partendo da un numero minimo di 3 (al di sotto devono esserci tutte). Per il secondo invece bastano 3 parole su 4, mentre la soglia minima è 4 (se le parole sono 3 devono esserci tutte).

Nel codice sopra è riportato l'esempio con il 75%. Questo viene realizzato tramite la divisione per 4 del numero totale di parole che compongono il KPI. Quindi si controlla se questa condizione è rispettata e nel caso in cui lo sia si incrementa il contatore delle occorrenze. Inoltre si calcola il numero di parole che sono state trovate e si memorizza questo dato.

4.1.4 Visualizzazione dei risultati

Una volta contate tutte le occorrenze di un'azienda, il software mostra i risultati. Per fare in modo che i dati siano facilmente leggibili, l'output verrà prodotto come file CSV. Il codice viene mostrato in **Figura 4.15**.

```
double totale = 0;
int count = 0;
File f = new File(a.getNome() + "_75%.csv");
FileWriter fw = new FileWriter(f);
fw.append("PROCESSO;KPI;REALE;TROVATO;% OVERLAP;OCCORRENZE;MEDIA\n");
for (Processo p : a.getProcessi()) {
    for (KPI kpi : p.getListaKPI()) {
        fw.append(p.getNome()); fw.append(";");
        fw.append(kpi.getNome()); fw.append(";");
        fw.append(""+kpi.getRilevazione()); fw.append(";");
        if (kpi.getOccorrenze() > 0) {
            fw.append("1");
        }
        else {
            fw.append("0");
        }
        fw.append(";;");
        fw.append(""+kpi.getOccorrenze()); fw.append(";");
        fw.append(""+kpi.getMedia()+"%"); fw.append(";\n");
    }
    fw.append(";;;;"+p.getOverlap()+";;\n");
    totale += p.getOverlap();
    count ++;
}
fw.append(";;;;;;\n");
fw.append("Risultati;;Media % overlap;;"+Utility.round(totale / count, 1)+";;\n");
fw.append(";;Media % match KPI;;" + calcolaMedia(a) + ";;\n");
fw.close();
```

Figura 4.15: codice per la stampa dei risultati.

Innanzitutto viene creato un file di tipo CSV con nome uguale a quello dell'azienda seguito dall'indicazione della percentuale usata nell'algoritmo.
La prima riga contiene la descrizione di ciascuna colonna, quindi ogni riga successiva contiene:

- Il nome del processo di appartenenza del KPI.
- Il nome del KPI.
- Il valore della misurazione reale.
- Un 1 o 0 a seconda che l'indicatore sia stato trovato o meno dall'algoritmo.
- Un valore percentuale che indica mediamente quante parole sono state trovate rispetto al totale delle parole che compongono il KPI stesso (calcolato sulle molteplici occorrenze).

Inoltre al termine di ogni processo viene mostrata la percentuale di "overlap" dei KPI che lo compongono: con overlap s'intende quanti valori corrispondono, quindi 1-1 e 0-0 sono esatti.

mentre 1-0 e 0-1 sono sbagliati. Il calcolo viene fatto tramite la funzione "getOverlap()" (**Figura 4.16**).

```
public double getOverlap() {
    double count = 0;
    int tot = 0;
    for (KPI k : listaKPI) {
        //Se la rilevazione è corretta aumenta count
        if ((k.getOccorrenze() == 0 && k.getRilevazione() == 0) ||
                (k.getOccorrenze() > 0 && k.getRilevazione() == 1))
            count++;

        tot++;
    }
    return Utility.round(count / tot * 100, 1);
}
```

Figura 4.16: codice del metodo "getOverlap()".

Infine viene calcolata la media di tutti gli overlap e anche la media dei match dei singoli KPI. La prima viene calcolata sommando i singoli overlap e dividendo per il numero totale, mentre la seconda tramite la funzione "calcolaMedia(Azienda a)" (**Figura 4.17**).

```
private static double calcolaMedia(Azienda a) {
    double res = 0;
    int count = 0;
    for (Processo p : a.getProcessi()) {
        for (KPI k : p.getListaKPI()) {
            if ((k.getOccorrenze() == 0 && k.getRilevazione() == 0) ||
                    (k.getOccorrenze() > 0 && k.getRilevazione() == 1))
                res++;

            count++;
        }
    }
    return Utility.round(res / count * 100, 1);
}
```

Figura 4.17: codice del metodo "calcolaMedia()".

Il software è perciò in grado di fornire subito un confronto con le misurazioni reali. Questo se e solo se sono stati inseriti i dati nel file contenente le rilevazioni. In **Figura 4.18** viene mostrato un esempio di output.

PROCESSO	KPI	REALE	TROVATO	% OVERLAP	MEDIA
CERTIFICAZIONI AMBIENTALI	ISO 14001	0	0		0.0%
CERTIFICAZIONI AMBIENTALI	EMAS	0	0		0.0%
CERTIFICAZIONI AMBIENTALI	ISO 50001	0	0		0.0%
CERTIFICAZIONI AMBIENTALI	ETICHETTA ECOLABEL	0	0		0.0%
CERTIFICAZIONI AMBIENTALI	LCA	0	1		100.0%
CERTIFICAZIONI AMBIENTALI	FSC	0	0		0.0%
CERTIFICAZIONI AMBIENTALI	GOLDPOWER	0	0		0.0%
CERTIFICAZIONI AMBIENTALI	LEED	0	0		0.0%
				87,50%	
CERTIFICAZIONI SOCIALI	SA 8000	0	0		0.0%
CERTIFICAZIONI SOCIALI	ISO 26000	0	0		0.0%
CERTIFICAZIONI SOCIALI	OHSAS 18001	0	0		0.0%
CERTIFICAZIONI SOCIALI	IFS	0	0		0.0%
CERTIFICAZIONI SOCIALI	ISO 22005	0	0		0.0%
CERTIFICAZIONI SOCIALI	ISO 22000	0	0		0.0%
				100,00%	
ENERGIA	UTILIZZO ENERGIE RINNOVABILI	0	1		100.0%
ENERGIA	POLITICHE DI RISPARMIO ENERGETICO	1	1		76.79%
ENERGIA	RIDUZIONE DEI COSTI	1	1		100.0%
ENERGIA	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		75.0%
ENERGIA	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	0	1		100.0%
ENERGIA	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	0		0.0%
ENERGIA	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO AL RISPARMIO DELLA RISORSA ENERGETICA	0	0		0.0%
				57,10%	
RISORSE PRIMARIE	E' PRESENTE UN IMPIANTO DI TRATTAMENTO/DEPURAZIONE/CAPTAZIONE DELLE ACQUE REFLUE	1	0		0.0%
RISORSE PRIMARIE	E' PREVISTO UN IMPIANTO DI TRATTAMENTO/DEPURAZIONE/CAPTAZIONE DELLE ACQUE PIOVANE	0	0		0.0%
RISORSE PRIMARIE	E' FORNITA INDICAZIONE CIRCA PERCENTUALE RIUTILIZZO ACQUE	0	0		0.0%
RISORSE PRIMARIE	E' PREVISTO L'UTILIZZO DI MATERIE PRIME RICICLATE PER PRODURRE I PROPRI PRODOTTI	0	1		76.92%
RISORSE PRIMARIE	E' PREVISTO UN RICICLO DEI PROPRI PRODOTTI FINITI FALLATI IN FASE DI SMALTIMENTO	0	1		76.92%
RISORSE PRIMARIE	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		75.0%
RISORSE PRIMARIE	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	0	0		0.0%
RISORSE PRIMARIE	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	0		0.0%
RISORSE PRIMARIE	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO AL RISPARMIO DELLA RISORSA IDRICA	0	0		0.0%
				55,60%	
GESTIONE RIFIUTI	SI FA RIFERIMENTO AD UNA NORMATIVA	0	1		84.62%
GESTIONE RIFIUTI	RACCOLTA DIFFERENZIATA: MODALITA' DI RACCOLTA (presente)	0	0		0.0%
GESTIONE RIFIUTI	ESISTE LA DESCRIZIONE DELLA MODALITA' DI RACCOLTA DIFFERENZIATA (VETRO, CARTA, ALLUMINIO...)	0	0		0.0%
GESTIONE RIFIUTI	E' PREVISTO IL RIUTILIZZO DEL RIFIUTO PER PRODUZIONE DI ELETTRICITA' E/O RISCALDAMENTO	0	0		0.0%
GESTIONE RIFIUTI	PACKAGING: VIENE UTILIZZATO MATERIALE BIODEGRADABILE(NO PLASTICA E DERIVATI)	1	0		0.0%
GESTIONE RIFIUTI	PACKAGING: RECUPERO E RIUTILIZZO	1	0		0.0%
GESTIONE RIFIUTI	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		75.0%

GESTIONE RIFIUTI	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA LA PRODUZIONE, RIDUZIONE O TRATTAMENTO DEI RIFIUTI	0	0		0.0%
GESTIONE RIFIUTI	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	0		0.0%
GESTIONE RIFIUTI	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO ALLA RIDUZIONE DI PRODUZIONE DEI RIFIUTI E/O ALL'AUMENTO DEL RICICLO DELLO STESSO	1	0		0.0%
				50,00%	
IMPATTO AMBIENTALE	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN ARIA	1	0		0.0%
IMPATTO AMBIENTALE	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN TERRA	0	0		0.0%
IMPATTO AMBIENTALE	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN ACQUA	0	0		0.0%
				66,70%	
REPORTING	E' PRESENTE IL BILANCIO DI SOSTENIBILITA'/ RAPPORTO AMBIENTALE/DICHIARAZIONE AMBIENTALE/BILANCIO AMBIENTALE E SOCIALE	0	0		0.0%
REPORTING	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		75.0%
REPORTING	SONO PRESENTI APPROFONDIMENTI DESCRITTIVI/TEMATICI	0	0		0.0%
				66,70%	
WELFARE	E' PRESENTE UNA POLITICA AZIENDALE DI PARI OPPORTUNITA'	0	0		0.0%
WELFARE	ASILO NIDO, FLESSIBILITA' ORARIA, ASISTENZA SANITARIA	0	0		0.0%
				100,00%	
RESPONSABILITA' SOCIALE	CODICE ETICO AZIENDALE	0	0		0.0%
RESPONSABILITA' SOCIALE	POLITICHE DI FORMAZIONE DEI PROPRI DIPENDENTI (eccetto quella obbligatoria)	0	0		0.0%
RESPONSABILITA' SOCIALE	E' PREVISTA LA VALUTAZIONE DI IMPATTO AMBIENTALE (VIA)	0	1		76.67%
RESPONSABILITA' SOCIALE	E' PREVISTA UNA POLITICA DI ANALISI DEL RISCHIO A TUTELA DEI DIPENDENTI NELL'AMBIENTE DI LAVORO	0	0		0.0%
				75,00%	
SUPPLY CHAIN	SONO PREVISTI CRITERI AMBIENTALI E SOCIALI DI SELEZIONE DI FORNITORI	0	0		0.0%
				100,00%	
VALORE PER IL CONSUMATORE	SONO PREVISTE POLITICHE DI INCENTIVAZIONE ALLA RESTITUZIONE IN AZIENDA DI PRODOTTI VECCHI/USURATI - CHE SCONTANO SULL'ACQUISTO DEL NUOVO	0	0		0.0%
VALORE PER IL CONSUMATORE	VIENE SPECIFICATA LA PROVENIENZA DELLE RISORSE E L'ORIGINE DI PRODUZIONE (TRASPARENZA DEL PRODOTTO E MADE IN ITALY)	0	0		0.0%
VALORE PER IL CONSUMATORE	ESISTONO POLITICHE DI COMUNICAZIONE SOCIALE SULLA SOSTENIBILITA' DEL PRODOTTO COMMERCIALIZZATO PER INFORMARE I PROPRI CONSUMATORI	1	0		0.0%
				66,70%	
Risultati		Media % overlap		75,00%	
		Media % match KPI		69,90%	

Figura 4.18: output dell'azienda Celli S.P.A.

4.2 Confronto dei risultati

Per verificare il grado di precisione dell'algoritmo sviluppato, si procede effettuando un'analisi dei dati ottenuti. Verranno presi in esame due casi limite, ovvero un'azienda con molti indicatori e una con meno. In **Figura 4.19** vengono mostrati i risultati per l'azienda Sacmi, mentre in **Figura 4.20** quelli di Celli.

Processo	KPI	Reale	ALG 50%	% overlap 1	ALG 75%	% overlap 2
CERTICAZIONI AMBIENTALI	ISO 14001	1	1		1	
	EMAS	0	1		1	
	ISO 50001	0	0		0	
	ETICHETTA ECOLABEL	0	0		0	
	LCA	0	1		1	
	FSC	0	1		0	
	GOLDPOWER	0	0		0	
	LEED	0	0	62,50%	0	75%
CERT SOCIALI	SA 8000	0	1		1	
	ISO 26000	0	0		0	
	OHSAS 18001	1	1		1	
	IFS	0	0		0	
	ISO 22005	0	0		0	
	ISO 22000	0	0	83%	0	83%
ENERGIA	UTILIZZO ENERGIE RINNOVABILI	1	1		1	
	POLITICHE DI RISPARMIO ENERGETICO	1	1		1	
	RIDUZIONE DEI COSTI	0	1		1	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	1	1		1	
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE...) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	1	1		1	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	1	1		0	
	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO AL RISPARMIO DELLA RISORSA ENERGETICA	0	1	71%	0	85%
RISORSE PRIMARIE	E' PRESENTE UN IMPIANTO DI TRATTAMENTO/DEPURAZIONE/CAPTAZIONE DELLE ACQUE REFLUE	0	1		0	
	E' PREVISTO UN IMPIANTO DI TRATTAMENTO/DEPURAZIONE/CAPTAZIONE DELLE ACQUE PIOVANE	0	1		0	
	E' FORNITA INDICAZIONE CIRCA PERCENTUALE RIUTILIZZO ACQUE	0	0		0	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	1	1		1	
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE...) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	1	1		0	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	1	1		0	
	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO AL RISPARMIO DELLA RISORSA IDRICA	0	1	55%	0	55%
GESTIONE RIFIUTI	SI FA RIFERIMENTO AD UNA NORMATIVA	0	1		1	
	RACCOLTA DIFFERENZIATA: MODALITA' DI RACCOLTA (presente)	1	1		1	
	ESISTE LA DESCRIZIONE DELLA MODALITA' DI RACCOLTA DIFFERENZIATA (VETRO, CARTA, ALLUMINIO..)	1	1		0	

	E' PREVISTO IL RIUTILIZZO DEL RIFIUTO PER PRODUZIONE DI ELETTRICITA' E/O RISCALDAMENTO	0	1		0	
	E' PREVISTO L'UTILIZZO DI MATERIE PRIME RICICLATE PER PRODURRE I PROPRI PRODOTTI	0	1		1	
	E' PREVISTO UN RICICLO DEI PROPRI PRODOTTI FINITI FALLATI IN FASE DI SMALTIMENTO	0	1		0	
	PACKAGING: VIENE UTILIZZATO MATERIALE BIODEGRADABILE(NO PLASTICA E DERIVATI)	0	1		1	
	PACKAGING: RECUPERO E RIUTILIZZO	0	1		1	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	1	1		1	
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE...) ANDAMENTI/TREND CIRCA LA PRODUZIONE, RIDUZIONE O TRATTAMENTO DEI RIFIUTI	1	1		0	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	1		0	
	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO ALLA RIDUZIONE DI PRODUZIONE DEI RIFIUTI E/O ALL'AUMENTO DEL RICICLO DELLO STESSO	0	1	40%	0	50%
IMPATTO AMBIENTALE	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN ARIA	1	1		0	
	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN TERRA	0	1		0	
	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN ACQUA	0	1	33%	0	66%
REPORTING	E' PRESENTE IL BILANCIO DI SOSTENIBILITA/ RAPPORTO AMBIENTALE/DICHIARAZIONE AMBIENTALE/BILANCIO AMBIENTALE E SOCIALE	1	1		1	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	1	1		1	
	SONO PRESENTI APPROFONDIMENTI DESCRITTIVI/TEMATICI	1	0	66%	0	66%
WELFARE	E' PRESENTE UNA POLITICA AZIENDALE DI PARI OPPORTUNITà	0	1		1	
	ASILO NIDO, FLESSIBILITà ORARIA, ASISTENZA SANITARIA	1	0	0%	0	0%
RESPONSABILITA' SOCIALE	CODICE ETICO AZIENDALE	1	1		1	
	POLITICHE DI FORMAZIONE DEI PROPRI DIPENDENTI (eccetto quella obbligatoria)	1	1		0	
	E' PREVISTA LA VALUTAZIONE DI IMPATTO AMBIENTALE (VIA)	0	1		1	
	E' PREVISTA UNA POLITICA DI ANALISI DEL RISCHIO A TUTELA DEI DIPENDENTI NELL'AMBIENTE DI LAVORO	1	1	75%	1	50%
SUPPLY CHAIN	SONO PREVISTI CRITERI AMBIENTALI E SOCIALI DI SELEZIONE DI FORNITORI	1	1	100%	0	0%
VALORE PER IL CONSUMATORE	SONO PREVISTE POLITICHE DI INCENTIVAZIONE ALLA RESTITUZIONE IN AZIENDA DI PRODOTTI VECCHI/USURATI - CHE SCONTANO SULL'ACQUISTO DEL NUOVO	0	1		0	
	VIENE SPECIFICATA LA PROVENIENZA DELLE RISORSE E L'ORIGINE DI PRODUZIONE (TRASPARENZA DEL PRODOTTO E MADE IN ITALY)	0	1		0	
	ESISTONO POLITICHE DI COMUNICAZIONE SOCIALE SULLA SOSTENIBILITA' DEL PRODOTTO COMMERCIALIZZATO PER INFORMARE I PROPRI CONSUMATORI	1	1	33%	0	66%
	Media % overlap			56%		54%
	Media % match singoli KPI			57%		66%

Figura 4.19: confronto dell'azienda Sacmi.

La terza colonna rappresenta il valore delle rilevazioni reali. La quarta e la sesta contengono i valori trovati dall'algoritmo, rispettivamente impostato con le soglie al 50% e al 75%. La quinta e la settima contengono le rispettive percentuali di overlap dei processi: per overlap s'intende quante rilevazioni del processo sono corrette, quindi 1-1 e 0-0 corrispondono a 1 mentre 1-0 o 0-1 a 0.

Al termine è riportata la media degli overlap per ciascuna soglia. L'algoritmo impostato al 50% da risultati migliori rispetto a quello al 75%, ma si tratta di un incremento marginale (da 54% a 56%) a scapito dell'utilizzo di una soglia molto più permissiva. Con questa percentuale, vengono trovati molti più KPI rispetto al secondo caso, che si traduce in un aumento dei match 1-1 ma anche di quelli 0-1: potrebbero essere trovati alcuni indicatori che in realtà non sono presenti.

Infine è riportata anche la media dei match dei singoli KPI. In questo caso è migliore quella al 75%. Questa media è più significativa rispetto alla precedente, in quanto rappresenta meglio il grado di precisione dell'algoritmo.

Da questo primo confronto emerge che l'algoritmo al 75% è probabilmente più accurato, in quanto evita di trovare indicatori che non sono realmente presenti per l'azienda.

Processo	KPI	Reale	ALG 50%	% overlap 1	ALG 75%	% overlap 2
CERTICAZIONI AMBIENTALI	ISO 14001	0	0		0	
	EMAS	0	0		0	
	ISO 50001	0	0		0	
	ETICHETTA ECOLABEL	0	0		0	
	LCA	0	1		1	
	FSC	0	0		0	
	GOLDPOWER	0	0		0	
	LEED	0	0	87,50%	0	87,50%
CERT SOCIALI	SA 8000	0	0		0	
	ISO 26000	0	0		0	
	OHSAS 18001	0	0		0	
	IFS	0	0		0	
	ISO 22005	0	0		0	
	ISO 22000	0	0	100%	0	100%
ENERGIA	UTILIZZO ENERGIE RINNOVABILI	0	1		1	
	POLITICHE DI RISPARMIO ENERGETICO	1	1		1	
	RIDUZIONE DEI COSTI	1	1		1	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		1	
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	0	1		1	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	1		0	
	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO AL RISPARMIO DELLA RISORSA ENERGETICA	0	1	28,50%	0	57,10%
RISORSE PRIMARIE	E' PRESENTE UN IMPIANTO DI TRATTAMENTO/DEPURAZIONE/CAPTAZIONE DELLE ACQUE REFLUE	1	1		0	
	E' PREVISTO UN IMPIANTO DI TRATTAMENTO/DEPURAZIONE/CAPTAZIONE DELLE ACQUE PIOVANE	0	1		0	
	E' FORNITA INDICAZIONE CIRCA PERCENTUALE RIUTILIZZO ACQUE	0	1		0	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		1	
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	0	1		0	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	1		0	
	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO AL RISPARMIO DELLA RISORSA IDRICA	0	1	14%	0	71%
GESTIONE RIFIUTI	SI FA RIFERIMENTO AD UNA NORMATIVA	0	1		1	
	RACCOLTA DIFFERENZIATA: MODALITA' DI RACCOLTA (presente)	0	0		0	
	ESISTE LA DESCRIZIONE DELLA MODALITA' DI RACCOLTA DIFFERENZIATA (VETRO, CARTA, ALLUMINIO..)	0	0		0	
	E' PREVISTO IL RIUTILIZZO DEL RIFIUTO PER PRODUZIONE DI ELETTRICITA' E/O RISCALDAMENTO	0	1		0	
	E' PREVISTO L'UTILIZZO DI MATERIE PRIME RICICLATE PER PRODURRE I PROPRI PRODOTTI	0	1		1	
	E' PREVISTO UN RICICLO DEI PROPRI PRODOTTI FINITI FALLATI IN FASE DI SMALTIMENTO	0	1		1	
	PACKAGING: VIENE UTILIZZATO MATERIALE BIODEGRADABILE(NO PLASTICA E DERIVATI)	1	0		0	
	PACKAGING: RECUPERO E RIUTILIZZO	1	1		0	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		1	
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA LA PRODUZIONE, RIDUZIONE O TRATTAMENTO DEI	0	0		0	

	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	1		0	
	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO ALLA RIDUZIONE DI PRODUZIONE DEI RIFIUTI E/O ALL'AUMENTO DEL RICICLO DELLO STESSO	1	1	45%	0	36%
IMPATTO AMBIENTALE	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN ARIA	1	1		0	
	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN TERRA	0	1		0	
	SONO MONITORATI GLI IMPATTI AMBIENTALI DERIVANTI DA EMISSIONI IN ACQUA	0	1	33%	0	66%
REPORTING	E' PRESENTE IL BILANCIO DI SOSTENIBILITA/ RAPPORTO AMBIENTALE/DICHIARAZIONE AMBIENTALE/BILANCIO AMBIENTALE E SOCIALE	0	1		0	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	1		1	
	SONO PRESENTI APPROFONDIMENTI DESCRITTIVI/TEMATICI	0	0	33%	0	66%
WELFARE	E' PRESENTE UNA POLITICA AZIENDALE DI PARI OPPORTUNITà	0	1		0	
	ASILO NIDO, FLESSIBILITà ORARIA, ASISTENZA SANITARIA	0	0	50%	0	100%
RESPONSABILITA' SOCIALE	CODICE ETICO AZIENDALE	0	1		0	
	POLITICHE DI FORMAZIONE DEI PROPRI DIPENDENTI (eccetto quella obbligatoria)	0	1		0	
	E' PREVISTA LA VALUTAZIONE DI IMPATTO AMBIENTALE (VIA)	0	1		1	
	E' PREVISTA UNA POLITICA DI ANALISI DEL RISCHIO A TUTELA DEI DIPENDENTI NELL'AMBIENTE DI LAVORO	0	1	0%	0	75%
SUPPLY CHAIN	SONO PREVISTI CRITERI AMBIENTALI E SOCIALI DI SELEZIONE DI FORNITORI	0	1	0%	0	100%
VALORE PER IL CONSUMATORE	SONO PREVISTE POLITICHE DI INCENTIVAZIONE ALLA RESTITUZIONE IN AZIENDA DI PRODOTTI VECCHI/USURATI - CHE SCONTANO SULL'ACQUISTO DEL NUOVO	0	1		0	
	VIENE SPECIFICATA LA PROVENIENZA DELLE RISORSE E L'ORIGINE DI PRODUZIONE (TRASPARENZA DEL PRODOTTO E MADE IN ITALY)	0	1		0	
	ESISTONO POLITICHE DI COMUNICAZIONE SOCIALE SULLA SOSTENIBILITA' DEL PRODOTTO COMMERCIALIZZATO PER INFORMARE I PROPRI CONSUMATORI	1	1	33%	0	66%
	Media % overlap			39%		75%
	Media % match singoli KPI			47%		70%

Figura 4.20: confronto dell'azienda Celli S.P.A.

Si nota subito leggendo la terza colonna che gli indicatori presenti sono meno rispetto a SACMI. Per questo motivo l'algoritmo al 50%, che tende a trovare tanti indicatori in quanto molto permissivo nella ricerca, ha una media decisamente più bassa rispetto alle misurazioni effettuate al 75%. Queste osservazioni valgono sia per la media degli overlap che per quella dei match dei singoli KPI. Perciò anche per Celli si conferma migliore l'algoritmo al 75%.

4.3 Analisi delle criticità

Durante la fase di download sono state riscontrate alcune difficoltà nello scaricamento dei siti. Innanzitutto alcuni siti sono interamente in inglese e perciò la ricerca dei KPI non darebbe risultati. Inoltre gli amministratori dei siti delle aziende possono negare il download ricorsivo delle pagine, principalmente per evitare sovraccarichi sui loro server. Infine alcuni siti sono troppo "pesanti" da scaricare: certi download possono richiedere ore e ore senza terminare, dovuto sia alla presenza di molte pagine e/o alle dimensioni delle stesse. Perciò alcune aziende potrebbero richiedere troppo tempo oppure non essere proprio analizzabili. Per questi problemi non c'è soluzione.
Molti siti scaricati presentavano pagine in più lingue (es. alcuni siti erano formati da metà pagine in inglese e metà in italiano). Dal momento che le pagine non in lingua italiana non porterebbero nessun match con i KPI, verranno cancellate prima di essere lette dal software in modo da alleggerire la quantità di dati da processare.

Nella fase di processing sorgono alcuni problemi legati all'algoritmo. In seguito ad un'analisi più accurata si nota che gli indicatori sono molto diversi tra di loro, alcuni sono semplici parole chiave, come le certificazioni ambientali e sociali (es. ISO14001 e IFS), altri sono composti da più parole ma rimangono "oggettivi" (es. riduzione dei costi) e i restanti sono invece vere e proprie frasi che spesso riguardano non solo dati testuali ma anche grafici (es. dettaglio trend: descrivere quale tipo di tendenze vengono illustrate e inserire il link). Inoltre alcuni indicatori non sono contestualizzati, ma sono generici e non significativi (es. sono previste azioni future nei prossimi 5 anni).
Per quanto riguarda i primi e i secondi si può migliorare introducendo un intervallo nella ricerca, mentre gli ultimi dovranno essere "semplificati" per renderli più adatti alla ricerca.

Capitolo 5 Riprogettazione, nuovo confronto e analisi delle performance

In questo capitolo verranno prima analizzate e poi implementate le modifiche all'algoritmo e al software stesso. Quindi verrà fatto un confronto prendendo un campione più ampio di aziende, suddivise per dimensioni.

5.1 Analisi delle modifiche

Tre sono le modifiche da effettuare: l'eliminazione delle pagine non in italiano, la "semplificazione" dei KPI e la definizione di intervalli di ricerca.

Per quanto concerne la prima, serviranno una funzione ricorsiva in grado di leggere tutto l'albero di pagine memorizzato all'interno della cartella contenente ciascun sito web e un metodo in grado di marchiare tutte le pagine che possiedono un contenuto in una lingua diversa da quella italiana.

Passando alla seconda, essa consiste nella riduzione delle parole che compongono i KPI, principalmente applicato a quelli più lunghi, che sono delle vere e proprie frasi. Infatti molti di questi indicatori non vengono quasi mai trovati con l'algoritmo al 75% nei casi mostrati in precedenza, ma sono invece spesso presenti al 50%: questo è dovuto al fatto che utilizzando una soglia più permissiva, è molto più probabile trovare un'occorrenza, ma spesso non corrisponde alla misurazione reale. Perciò l'algoritmo al 50% verrà scartato e si utilizzerà solo l'altro.

L'obiettivo è quello di ridurre i KPI lasciando poche parole chiave. Il primo passo è togliere dai KPI tutte le parole che sono significative per la ricerca, come ad esempio articoli e preposizioni, ma anche i verbi, che potrebbero comparire in forme diverse. Quindi si procede modificando i singolari/plurali delle parole per cercare di rendere gli indicatori più simili a come potrebbero apparire nei siti. Per applicare la modifica basterà cambiare il file contenente i KPI.

Infine per l'ultimo aspetto si deve pensare a come l'algoritmo cerca i KPI: verifica la presenza delle parole che lo compongono l'indicatore all'interno di tutta la pagina. Questo significa che potrebbe trovare le singole parole in parti diverse (quindi in più paragrafi e contesti) e/o lontane (ad esempio potrebbe trovare una parola all'inizio e una alla fine della pagina, quindi il KPI verrebbe trovato ma con un "criterio" non corretto).

Verrà perciò introdotto un intervallo in cui cercare le parole che compongono il KPI: nel momento in cui ne viene trovata una, il software limiterà la ricerca alle parole limitrofe. Ciascun KPI avrà il proprio intervallo, ma sarà comunque possibile avere indicatori senza questa modifica. Il valore dell'intervallo dipenderà principalmente dalla lunghezza del KPI: generalmente più saranno le parole che lo compongono più sarà grande e viceversa. In secondo piano verrà fatta un'analisi soggettiva per ciascun KPI, cercando di definire intervalli che rispecchino la possibile disposizione delle parole nei siti.

Indicatori come le certificazioni (es. ISO 14001 e SA 8000) avranno un intervallo pari a 2, cioè trovata una delle due parole l'altra dovrà trovarsi adiacente alla prima.

In questo caso per applicare il cambiamento bisognerà manipolare ulteriormente il file dei KPI, aggiungendo un valore per l'intervallo. Il software verrà modificato sia nella prima parte, dove viene letto il file, sia nella terza, dove viene eseguito l'algoritmo.

5.2 Implementazione delle modifiche

Per quanto riguarda i cambiamenti al file dei KPI, esso risulterà come mostrato in **Figura 5.1**.

Processo	KPI	Intervallo	KPI semplificato	Additional keywords	
ENERGIA	UTILIZZO ENERGIE RINNOVABILI	20	ENERGIE RINNOVABILI	pale eoliche	mulino ad acqua
	POLITICHE DI RISPARMIO ENERGETICO	50	POLITICHE RISPARMIO ENERGETICO	lampade a risparmio energetico	cappotto esterno
	RIDUZIONE DEI COSTI	0	RIDUZIONE COSTI ENERGIA	Indicazione di minori costi e risparmi economici dovuti ad utilizzo di energie rinnovabili	
	SONO PREVISTE AZIONI FUTURE NEI PROSSIMI 5 ANNI	0	AZIONI FUTURE 5 ANNI RIDUZIONE ENERGIA		
	VENGONO RAPPRESENTATI (CON GRAFICI O TABELLE…) ANDAMENTI/TREND CIRCA L'UTILIZZO O LA RIDUZIONE DI UTILIZZO DELLA RISORSA	0	GRAFICI UTILIZZO RIDUZIONE ENERGIA	consumo energia	utilizzo energia
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0			
	SONO PREVISTE AZIONI/POLITICHE DI SENSIBILIZZAZIONE IN MERITO AL RISPARMIO DELLA RISORSA ENERGETICA	200	POLITICHE SENSIBILIZZAZIONE RISPARMIO ENERGETICO		

Figura 5.1: file contenente alcuni dei KPI in seguito alle modifiche.

Per quanto riguarda il software verranno analizzate singolarmente le tre parti sottoposte a cambiamenti.

1) Lettura delle pagine

La fase di lettura è stata modificata come mostrata in **Figura 5.2**.

```
// Cerca la directory che contiene le pagine
for (File f : siteList) {
    if (f.isDirectory() && a.getSito().contains(f.getName())) {
        PageReader.cancellaPagineNonItaliane(f.getAbsolutePath());
        // Lettura ricorsiva delle pagine data la directory principale
        PageReader.leggiPagine(a, f.getAbsolutePath());
    }
}
```

Figura 5.2: codice per la cancellazione e lettura delle pagine.

È stata aggiunta la funzione ricorsiva "cancellaPagineNonItaliane(String fileName)" (**Figura 5.3**), che prende come parametro il percorso assoluto della cartella contenente i file. Essa è in grado di eliminare tutte le pagine che hanno il contenuto non in lingua italiana.

```
public static void cancellaPagineNonItaliane(String fileName) {
    File file = new File(fileName);
    //Se è una directory, riesegue la funzione per tutti i file che trova
    if (file.isDirectory()) {
        File[] fileList = file.listFiles();
        for (File f : fileList) {
            cancellaPagineNonItaliane(f.getAbsolutePath());
        }
    }
    else {
        if (file.isFile() && file.getName().endsWith(".html")) {
            //Se il file è HTML e la pagina non è italiana lo cancella
            if (!HtmlParser.isPageItalian(file.getAbsolutePath())) {
                file.delete();
            }
        }
    }
}
```

Figura 5.3: codice del metodo "cancellaPagineNonItaliane()".

Il metodo crea un oggetto File con il nome passato e controlla che sia una directory:

- Se è vero vengono letti tutti i file contenuti nella cartella e per ciascuno viene invocata nuovamente la “cancellaPagineNonItaliane()”.
- Se è falso controlla che sia un file e che l’estensione sia “.html”: se entrambe sono vere invoca la funzione “isPageItalian(String fileName)”.

Se “isPageItalian(String fileName)” ritorna false, ovvero la pagina non è italiana, allora elimina il file, altrimenti non fa nulla. Il metodo viene mostrato in **Figura 5.4**.

```
public static boolean isPageItalian(String fileName) {
    boolean res = false;
    try {
        File input = new File(fileName);
        Document doc = Jsoup.parse(input, "UTF-8");
        //Seleziona il tag html
        Element langTag = doc.select("html").first();
        //Cerca l'attributo lang
        if (!langTag.attr("lang").equals("")) {
            //Se è diverso da it ritorna true
            if (langTag.attr("lang").toLowerCase().contains("it"))
                res = true;
        }
        else {
            res = true;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return res;
}
```

Figura 5.4: codice del metodo “isPageItalian()”.

La funzione “isPageItalian(String fileName)” verifica se il contenuto è in lingua italiana. Innanzitutto legge il file parsificandolo come se dovesse leggerlo. Quindi estrae il tag <html> e in particolare da esso l’attributo “lang”, che contiene la lingua utilizzata nel documento. Se l’attributo non esiste ritorna true, in quanto non è possibile determinare la lingua. Se esiste ed è uguale ad “it”, ovvero italiano, restituisce sempre true. In ogni altro caso ritorna false.

2) Lettura dei KPI

Il metodo "leggiKPI(String fileName, Azienda a)" viene modificato come mostrato in **Figura 5.5**.

```
public static void leggiKPI(String fileName, Azienda a) {
    //Mappa ogni processo con una lista di KPI
    //Ogni KPI a sua volta contiene un nome e una lista di additional keywords
    try {
        //Parsing del file
        CSVReader reader = new CSVReader(new FileReader(fileName), ';');
        //Scarta la prima riga che contiene i nomi delle colonne
        reader.readNext();

        String[] nextLine;
        Processo processo = null;
        while ((nextLine = reader.readNext()) != null) {
            //Se trova un nuovo processo, crea una un nuovo oggetto
            if (!nextLine[0].equals("")) {
                processo = new Processo(nextLine[0]);
                a.getProcessi().add(processo);
            }
            //Legge il kpi
            KPI newKpi;
            if (nextLine[2].equals(""))
                newKpi = new KPI(nextLine[1], Integer.parseInt(nextLine[3]));
            else
                newKpi = new KPI(nextLine[2], Integer.parseInt(nextLine[3]));
            //Aggiunge le additional keywords al kpi
            for (int i=4; i<nextLine.length; i++) {
                if (!nextLine[i].equals("")) {
                    newKpi.aggiungiKeyword(nextLine[i]);
                }
            }
            //Inserisce il kpi nella lista del processo
            processo.getListaKPI().add(newKpi);
        }
        //Chiude il reader
        reader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 5.5: codice del metodo "leggiKPI()".

Innanzitutto il costruttore del KPI richiede anche un secondo parametro che rappresenta l'intervallo. Il nome sarà quello semplificato se esiste, altrimenti quello base. Il ciclo delle parole addizionali inizierà con indice 4 anziché 2, dovuto alla presenza di due ulteriori colonne.

3) Processing e Algoritmo

Il metodo "contaOccorrenze(List<String> pages)" si modifica come mostrato in **Figura 5.6**.

```
public void contaOccorrenze(List<String> pages) {
    if (intervallo == 0) {
        for (String page : pages) {
            conta(page);
        }
    }
    else {
        for (String page : pages) {
            contaIntervallo(page);
        }
    }
}
```

Figura 5.6: codice del metodo "contaOccorrenze()".

Per distinguere se la ricerca del KPI deve essere ristretta o meno, si agisce sul valore dell'intervallo: se questo viene impostato a 0 la ricerca viene fatta in tutta la pagina, se invece viene impostato ad un qualunque numero positivo, la ricerca viene effettuata nell'intorno specificato.

Prendendo come esempio il KPI "politiche risparmio energetico", esso ha un intervallo pari a 50: questo significa che trovando una delle parole, le altre vengono cercate prendendo un intervallo di 50 parole centrato su quella trovata. Quindi se viene trovata "politiche" in posizione 100, la ricerca delle restanti viene fatta tra le posizioni 75 e 125.

La variabile "intervallo" ne contiene il valore (diviso per due quando memorizzato): se vale 0 viene invocata la "conta(String page)" che è la stessa mostrata nel Capitolo 4, altrimenti la "contaIntervallo(String page)" (**Figura 5.7**), che contiene il nuovo algoritmo.

```
private void contaIntervallo(String page) {
    String regexp = "[^\\p{L}\\p{Nd}]+";
    //Divide la pagina nella singole parole
    List<String> parole = Arrays.asList(page.split(regexp));

    for (String key : keywords) {
        //Separa la key nelle singole parole, usando la regexp
        List<String> temp = Arrays.asList(key.toLowerCase().split(regexp));
        double count = 0;   //Contatore delle parole trovate
        double words = temp.size(); //Memorizza il totale delle parole
        int posizione = -1;
        for (int i=0; i<temp.size(); i++) {
            for (int j=0; j<parole.size(); j++) {
                if (parole.get(j).equals(temp.get(i))) {
                    posizione = j;
                    break;
                }
            }
        }

        if (posizione >= 0) {
            int min = 0;
            int max = parole.size();
            String newPagina = "";
            if ((posizione - intervallo) > min)
                min = posizione - intervallo;
            if ((posizione + intervallo) < max)
                max = posizione + intervallo;
            List<String> newParole = parole.subList(min, max);
            for (String s : newParole)
                newPagina += s;
            for (int i=0; i<temp.size(); i++) {
                if (newPagina.contains(temp.get(i))) {
                    count++;
                }
            }

            int def = temp.size() / 4;  //Approssimazione al 75%
            if (words >= 4)             //Ogni 4 parole ne "toglie" 1
                words = words - def;    //Se key ha 8 parole ne bastano 6
            if (count >= words) {
                occorrenze++;
                //Calcola quante parole sono state trovate e lo memorizza
                double precisione = count / temp.size();
                precisioni.put(key, new Double(precisione));
                totale += precisione;
            }
        }
    }
}
```

Figura 5.7: codice del metodo "contaIntervallo()".

Per prima cosa è necessario spezzare anche la pagina nelle singole stringhe che la compongono, sempre tramite la regexp definita in precedenza e la funzione “split()”.
Il metodo definisce una variabile posizione che serve per memorizzare l’indice della prima parola trovata di quelle che compongono il KPI. Viene inizializzata a -1.
Tramite il ciclo vengono cercate le parole all’interno della pagina (o meglio, in tutte le stringhe che la compongono): quando ne viene trovata una, viene memorizzato l’indice. Il break serve per uscire immediatamente dal ciclo e passare alla parte successiva.
Se al termine non sono state trovate parole, posizione vale -1, il controllo restituisce falso e non vengono eseguite ulteriori operazioni.
Se è stata trovata una parola, la condizione del controllo è vera e si prosegue con la creazione di una sotto-lista contenente solo le parole presenti nell’intervallo definito. È necessaria molta attenzione nella realizzazione dell’intervallo: si potrebbe infatti finire fuori scala dalla lista.
Una lista infatti inizia con indice 0 e termina con indice pari al totale degli elementi contenuti meno uno. Se ad esempio la posizione vale 10 e l’intervallo del KPI è 50, l’indice di partenza diventerebbe -40 e sarebbe fuori scala. Lo stesso vale per l’indice massimo.
Il metodo controlla che nessuno dei due sfori, altrimenti imposta come minimo 0 e come massimo la dimensione della lista. Quindi procede creando la sotto-lista e riunisce tutte le parole contenute in un'unica stringa. A questo punto esegue un ciclo dove cerca le parole che compongono il KPI dentro la stringa appena creata (ovvero la parte di pagina definita dall’intervallo) e ogni volta che ne trova una incrementa il contatore.
Infine esegue le operazioni già definite nella prima implementazione: utilizza l’algoritmo al 75% come soglia, quindi verifica se sono state trovate sufficienti parole e in caso positivo incrementa le occorrenze, calcolando sempre la percentuale di parole trovate.

5.3 Visualizzazione dei risultati

Vengono mostrati i risultati ottenuti in seguito alle modifiche, utilizzando solamente l’algoritmo al 75%. In **Figura 5.8** vengono mostrati i nuovi risultati dell’azienda Sacmi, mentre in **Figura 5.9** quelli di Celli.

Processo	KPI	Reale	ALG 75%	% overlap
CERTIFICAZIONI AMBIENTALI	ISO 14001	1	1	
	EMAS	0	0	
	ISO 50001	0	0	
	ETICHETTA ECOLABEL	0	0	
	LCA	0	0	
	FSC	0	0	
	GOLDPOWER	0	0	
	LEED	0	0	100%
CERTIFICAZIONI SOCIALI	SA 8000	0	0	
	ISO 26000	0	0	
	OHSAS 18001	1	1	
	IFS	0	0	
	ISO 22005	0	0	
	ISO 22000	0	0	100%
ENERGIA	ENERGIE RINNOVABILI	1	0	
	POLITICHE RISPARMIO ENERGETICO	1	1	
	RIDUZIONE COSTI ENERGIA	0	1	
	AZIONI FUTURE 5 ANNI RIDUZIONE ENERGIA	1	1	
	GRAFICI UTILIZZO RIDUZIONE ENERGIA	1	1	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	1	0	
	POLITICHE SENSIBILIZZAZIONE RISPARMIO ENERGETICO	0	0	57,10%
RISORSE PRIMARIE	IMPIANTO TRATTAMENTO DEPURAZIONE CAPTAZIONE ACQUE REFLUE	0	0	
	IMPIANTO TRATTAMENTO DEPURAZIONE CAPTAZIONE ACQUE PIOVANE	0	0	
	PERCENTUALE RIUTILIZZO ACQUE	0	0	
	AZIONI FUTURE 5 ANNI RISORSE PRIMARIE	1	0	
	GRAFICI RISORSE PRIMARIE	1	0	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	1	0	
	POLITICHE SENSIBILIZZAZIONE RISPARMIO RISORSA IDRICA	0	0	57,10%
GESTIONE RIFIUTI	NORMATIVA GESTIONE RIFIUTI	0	0	
	RACCOLTA DIFFERENZIATA	1	0	
	DESCRIZIONE RACCOLTA DIFFERENZIATA	1	1	
	RIUTILIZZO RIFIUTI RISCALDAMENTO ELETTRICITà	0	0	
	UTILIZZO MATERIE PRIME RICICLATE PRODOTTI	0	1	
	RICICLO PRODOTTI FINITI FALLATI	0	0	
	UTILIZZO MATERIALE BIODEGRADABILE	0	0	
	RECUPERO/RIUTILIZZO PACKAGING	0	0	
	AZIONI FUTURE 5 ANNI GESTIONE RIFIUTI	1	1	
	GRAFICI RIDUZIONE TRATTAMENTO RIFIUTI	1	0	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	0	
	POLITICHE SENSIBILIZZAZIONE RICICLO	0	0	75%
IMPATTO AMBIENTALE	IMPATTO AMBIENTALE EMISSIONI ARIA	1	0	
	IMPATTO AMBIENTALE EMISSIONI TERRA	0	0	
	IMPATTO AMBIENTALE EMISSIONI ACQUA	0	0	66,70%
REPORTING	BILANCIO SOSTENIBILITà RAPPORTO AMBIENTALE	1	1	
	AZIONI FUTURE 5 ANNI RAPPORTO AMBIENTALE	1	1	
	APPROFONDIMENTI AMBIENTALI	1	0	66,70%

WELFARE	POLITICA AZIENDALE PARI OPPORTUNITà	0	0	
	ASILO NIDO, FLESSIBILITà ORARIA, ASISTENZA SANITARIA	1	0	50%
RESPONSABILITA' SOCIALE	CODICE ETICO AZIENDALE	1	1	
	POLITICHE FORMAZIONE DIPENDENTI	1	0	
	VALUTAZIONE IMPATTO AMBIENTALE	0	0	
	POLITICA ANALISI RISCHI TUTELA DIPENDENTI AMBIENTE LAVORO	1	1	75%
SUPPLY CHAIN	CRITERI AMBIENTALI SOCIALI SELEZIONE FORNITORI	1	0	0%
VALORE PER IL CONSUMATORE	POLITICHE INCENTIVAZIONE RESTITUZIONE PRODOTTI	0	0	
	PROVENIENZA RISORSE ORIGINE PRODUZIONE	0	0	
	COMUNICAZIONE SOSTENIBILITà PRODOTTI	1	0	66,70%
	Media % overlap			64,94%
	Media % match KPI			73,20%

Figura 5.8: confronto dell'azienda Sacmi in seguito alle modifiche.

Per l'azienda Sacmi i risultati sono migliorati. La media overlap è passata da 54% a 64.9%, mentre la media dei match KPI da 66% a 73.2%.

Processo	KPI	Reale	ALG 75%	% overlap
CERTIFICAZIONI AMBIENTALI	ISO 14001	0	0	
	EMAS	0	0	
	ISO 50001	0	0	
	ETICHETTA ECOLABEL	0	0	
	LCA	0	1	
	FSC	0	0	
	GOLDPOWER	0	0	
	LEED	0	0	87,50%
CERTIFICAZIONI SOCIALI	SA 8000	0	0	
	ISO 26000	0	0	
	OHSAS 18001	0	0	
	IFS	0	0	
	ISO 22005	0	0	
	ISO 22000	0	0	100%
ENERGIA	ENERGIE RINNOVABILI	0	1	
	POLITICHE RISPARMIO ENERGETICO	1	1	
	RIDUZIONE COSTI ENERGIA	1	1	
	AZIONI FUTURE 5 ANNI RIDUZIONE ENERGIA	0	1	
	GRAFICI UTILIZZO RIDUZIONE ENERGIA	0	1	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	0	
	POLITICHE SENSIBILIZZAZIONE RISPARMIO ENERGETICO	0	0	57,10%
RISORSE PRIMARIE	IMPIANTO TRATTAMENTO DEPURAZIONE CAPTAZIONE ACQUE REFLUE	1	0	
	PIOVANE	0	0	
	PERCENTUALE RIUTILIZZO ACQUE	0	0	
	AZIONI FUTURE 5 ANNI RISORSE PRIMARIE	0	0	
	GRAFICI RISORSE PRIMARIE	0	0	

	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	0	
	POLITICHE SENSIBILIZZAZIONE RISPARMIO RISORSA IDRICA	0	0	85,70%
GESTIONE RIFIUTI	NORMATIVA GESTIONE RIFIUTI	0	0	
	RACCOLTA DIFFERENZIATA	0	0	
	DESCRIZIONE RACCOLTA DIFFERENZIATA	0	0	
	RIUTILIZZO RIFIUTI RISCALDAMENTO ELETTRICITà	0	0	
	UTILIZZO MATERIE PRIME RICICLATE PRODOTTI	0	1	
	RICICLO PRODOTTI FINITI FALLATI	0	0	
	UTILIZZO MATERIALE BIODEGRADABILE	1	0	
	RECUPERO/RIUTILIZZO PACKAGING	1	0	
	AZIONI FUTURE 5 ANNI GESTIONE RIFIUTI	0	1	
	GRAFICI RIDUZIONE TRATTAMENTO RIFIUTI	0	0	
	DETTAGLIO TREND: DESCRIVERE QUALE TIPO DI TENDENZE VENGONO ILLUSTRATE E INSERIRE IL LINK	0	0	
	POLITICHE SENSIBILIZZAZIONE RICICLO	1	0	58,30%
IMPATTO AMBIENTALE	IMPATTO AMBIENTALE EMISSIONI ARIA	1	0	
	IMPATTO AMBIENTALE EMISSIONI TERRA	0	0	
	IMPATTO AMBIENTALE EMISSIONI ACQUA	0	0	66,70%
REPORTING	BILANCIO SOSTENIBILITà RAPPORTO AMBIENTALE	0	0	
	AZIONI FUTURE 5 ANNI RAPPORTO AMBIENTALE	0	1	
	APPROFONDIMENTI AMBIENTALI	0	0	66,70%
WELFARE	POLITICA AZIENDALE PARI OPPORTUNITà	0	0	
	ASILO NIDO, FLESSIBILITà ORARIA, ASISTENZA SANITARIA	0	0	100%
RESPONSABILITA' SOCIALE	CODICE ETICO AZIENDALE	0	0	
	POLITICHE FORMAZIONE DIPENDENTI	0	0	
	VALUTAZIONE IMPATTO AMBIENTALE	0	0	
	POLITICA ANALISI RISCHI TUTELA DIPENDENTI AMBIENTE LAVORO	0	0	100%
SUPPLY CHAIN	CRITERI AMBIENTALI SOCIALI SELEZIONE FORNITORI	0	0	100%
VALORE PER IL CONSUMATORE	POLITICHE INCENTIVAZIONE RESTITUZIONE PRODOTTI	0	0	
	PROVENIENZA RISORSE ORIGINE PRODUZIONE	0	1	
	COMUNICAZIONE SOSTENIBILITà PRODOTTI	1	1	66,70%
	Media % overlap			80,79%
	Media % match KPI			76,80%

Figura 5.9: confronto dell'azienda Celli in seguito alle modifiche.

Anche per Celli i risultati sono migliorati. La media overlap è passata da 75% a 80.8%, mentre la media dei match KPI da 70% a 76.8%.

L'introduzione degli intervalli ha permesso di rendere quasi perfetti i KPI dei processi "certificazioni" e ha migliorato quelli composti da 2-3 parole.
Per quanto riguarda gli indicatori più lunghi, nonostante siano stati ridotti e resi più adatti alla ricerca, la situazione è solo leggermente migliorata: i match non sempre sono precisi, a volte i KPI vengono trovati anche se non realmente presenti secondo le misurazioni reali.

5.4 Confronto finale

Viene eseguito ora un confronto prendendo un campione di 30 aziende selezionate dalla lista delle 100. Non è stato possibile considerare un numero maggiore di aziende principalmente per i problemi legati al download discussi nel paragrafo 4.3.
La scelta è stata fatta principalmente scegliendo aziende che possedevano un numero elevato di indicatori. Inoltre è stata fatta una classificazione in base al numero dei dipendenti, definendo tre fasce e prendendo per ciascuna 10 aziende:

- >=300 dipendenti
- >=200 e <300 dipendenti
- <200 dipendenti

Le tabelle in **Figura 5.10**, **Figura 5.11** e **Figura 5.12** mostrano le medie % di overlap e di match KPI delle aziende suddivise nelle tre fasce.

Azienda	N° dipendenti	% overlap	% match KPI
IMA S.P.A.	1357	40,00%	46,40%
Argo Tractors S.P.A.	1248	74,60%	73,20%
BONFIGLIOLI RIDUTTORI S.P.A.	1240	71,20%	69,60%
SACMI IMOLA S.C.	1085	64,90%	73,20%
CORGHI S.P.A.	684	86,80%	89,30%
BREVINI POWER TRANSMISSION S.P.A.	617	82,10%	83,90%
SYSTEM S.P.A.	508	76,90%	71,40%
Hydrocontrol S.P.A.	418	100,00%	100,00%
MARINI S.P.A.	377	75,80%	71,40%
Interpump Group S.P.A.	362	91,30%	89,30%
Medie aziende >300 dipendenti		**76,36%**	**76,77%**

Figura 5.10: tabella delle aziende con >300 dipendenti.

Azienda	N° dipendenti	% overlap	% match KPI
WITTUR S.P.A.	291	86,50%	85,70%
Gruppo Fabbri Vignola S.P.A.	284	77,90%	83,90%
Mec-Track S.R.L.	268	51,10%	66,10%
MODULA S.P.A.	241	87,00%	89,30%
COMECER S.P.A.	237	61,80%	69,60%
BOLZONI S.P.A.	235	89,40%	92,90%
ACMI S.P.A.	230	91,30%	91,10%
Saer Elettropompe S.P.A.	214	92,60%	92,90%
Dinamic Oil S.P.A.	203	97,40%	96,40%
NORDMECCANICA S.P.A.	200	98,70%	98,20%
Medie aziende >=200 e <300 dipendenti		**83,37%**	**86,61%**

Figura 5.11: tabella delle aziende con >=200 e <300 dipendenti.

Azienda	N° dipendenti	% overlap	% match KPI
O.M.P. - OFFICINE MAZZOCCO PAGNONI - S.R.L.	187	84,30%	73,20%
DULEVO INTERNATIONAL S.P.A.	187	87,60%	89,30%
SOCIETA' PER AZIONI CURTI	164	84,70%	85,70%
IMAL S.R.L.	162	89,30%	89,30%
Terex Italia S.R.L.	148	78,80%	91,10%
Bilfinger Water Technologies S.R.L.	123	50,60%	60,70%
Celli S.P.A.	107	80,80%	76,80%
WERTHER INTERNATIONAL S.P.A.	96	100,00%	100,00%
In. Te. Sa. S.P.A.	90	89,80%	89,30%
Fiori Group S.P.A.	72	92,10%	92,90%
Medie aziende <200 dipendenti		**83,80%**	**84,83%**

Figura 5.12: tabella delle aziende con <200 dipendenti.

Da questa prima analisi emergono alcune considerazioni:

- La media % dei match KPI è sempre migliore rispetto a quella di overlap.
- Le aziende più grandi hanno percentuali mediamente più basse: probabilmente è dovuto al maggior numero di indicatori posseduti.

Per verificare la seconda considerazione, viene fatto un nuovo confronto utilizzando appunto una suddivisione in base al numero di indicatori posseduti, sempre in 3 fasce:

- >=10 indicatori
- >=5 e <10 indicatori
- <5 indicatori

Nelle tabelle di **Figura 5.13**, **Figura 5.14** e **Figura 5.15** sono riportate le aziende ordinate per numero di indicatori.

Azienda	N° dipendenti	N° indicatori	% overlap	% match KPI
IMA S.P.A.	1357	28	40,00%	46,40%
Mec-Track S.R.L.	268	29	51,10%	66,10%
Bilfinger Water Technologies S.R.L.	123	26	50,60%	60,70%
SACMI IMOLA S.C.	1085	24	64,90%	73,20%
Argo Tractors S.P.A.	1248	20	74,60%	73,20%
O.M.P. - OFFICINE MAZZOCCO PAGNONI - S.R.L.	187	18	84,30%	73,20%
MARINI S.P.A.	377	17	75,80%	71,40%
COMECER S.P.A.	237	16	61,80%	69,60%
Gruppo Fabbri Vignola S.P.A.	284	10	77,90%	83,90%
Medie aziende >=10 indicatori			**64,56%**	**68,63%**

Figura 5.13: tabella delle aziende con >=10 indicatori.

Azienda	N° dipendenti	N° indicatori	% overlap	% match KPI
SYSTEM S.P.A.	508	9	76,90%	71,40%
Celli S.P.A.	107	8	80,80%	76,80%
BONFIGLIOLI RIDUTTORI S.P.A.	1240	7	71,20%	69,60%
SOCIETA' PER AZIONI CURTI	164	7	84,70%	85,70%
BREVINI POWER TRANSMISSION S.P.A.	617	6	82,10%	83,90%
MODULA S.P.A.	241	5	87,00%	89,30%
Terex Italia S.R.L.	148	5	78,80%	91,10%
Fiori Group S.P.A.	72	5	92,10%	92,90%
Medie aziende >=5 e <10 indicatori			**81,70%**	**82,59%**

Figura 5.14: tabella delle aziende con >=5 e <10 indicatori.

Azienda	N° dipendenti	N° indicatori	% overlap	% match KPI
BOLZONI S.P.A.	235	4	89,40%	92,90%
Saer Elettropompe S.P.A.	214	3	92,60%	92,90%
In. Te. Sa. S.P.A.	90	3	89,80%	89,30%
Interpump Group S.P.A.	362	2	91,30%	89,30%
CORGHI S.P.A.	684	1	86,80%	89,30%
Hydrocontrol S.P.A.	418	1	100,00%	100,00%
WITTUR S.P.A.	291	1	86,50%	85,70%
DULEVO INTERNATIONAL S.P.A.	187	1	87,60%	89,30%
ACMI S.P.A.	230	0	91,30%	91,10%
Dinamic Oil S.P.A.	203	0	97,40%	96,40%
NORDMECCANICA S.P.A.	200	0	98,70%	98,20%
IMAL S.R.L.	162	0	89,30%	89,30%
WERTHER INTERNATIONAL S.P.A.	96	0	100,00%	100,00%
Medie aziende <5 indicatori			**92,36%**	**92,59%**

Figura 5.15: tabella delle aziende con <5 indicatori.

Dalla seconda analisi emergono considerazioni più rilevanti:

- Come previsto, le aziende con più indicatori hanno percentuali più basse. Le medie sono comunque ottime tranne due eccezioni, Ima e Bilfinger.
- La seconda e la terza fascia mostrano risultati molti buoni. Nonostante siano meno rilevanti in quanto hanno generalmente pochi indicatori, da un altro punto di vista è comunque importante perché significa che l'algoritmo tende a non trovare indicatori che non sono realmente presenti.

Si è cercato di indagare per capire i motivi che hanno portato ad avere due aziende con percentuali così basse rispetto alla media:

- Ima: analizzando il sito web si è visto che molte informazioni si trovavano all'interno di documenti pdf, elementi non trattati in questa tesi.
- Bilfinger: analizzando il sito e le pagine scaricate si è notato che la maggior parte erano in inglese e solo poche in italiano, limitando perciò molto la ricerca.

Togliendo queste due aziende da quelle con >=10 indicatori, i risultati migliorano come mostrato in **Figura 5.16**.

Medie aziende >=10 indicatori	% overlap	% match KPI
Prima	64,56%	68,63%
Dopo	**70,06%**	**72,94%**

Figura 5.16: medie delle aziende con >=10 indicatori togliendo i due casi particolari.

A questo punto si possono riassumere i risultati ottenuti per le tre fasce e si ottengono dei valori molto più lineari: si ottiene uno scarto di circa il 10% da una fascia all'altra, come si vede in **Figura 5.17**.

Categoria	% overlap	% match KPI
Medie aziende >=10 indicatori	70,06%	72,94%
Medie aziende >=5 e <10 indicatc	81,70%	82,59%
Medie aziende <5 indicatori	92,36%	92,59%

Figura 5.17: medie finali delle tre fasce di aziende.

I risultati ottenuti sono perciò molto soddisfacenti.

5.5 Analisi delle performance

Verranno analizzate le performance del software durante il processamento delle 30 aziende prima trattate. Non verrà considerato il tempo richiesto per il download dei siti, in quanto può essere anche molto lungo. Inoltre può dipendere da molti fattori, principalmente dalla velocità della connessione e dal server da cui si scarica.

La macchina utilizzata per il test dispone di un processore a quattro core e 16 GB di RAM. Il tempo totale richiesto per l'esecuzione è stato di 1:48 minuti. Nei grafici di **Figura 5.18** e **Figura 5.19** sono riportati rispettivamente l'impiego della memoria RAM e l'utilizzo della CPU. Il monitoraggio è stato effettuato tramite JConsole, un tool presente all'interno della suite Java.

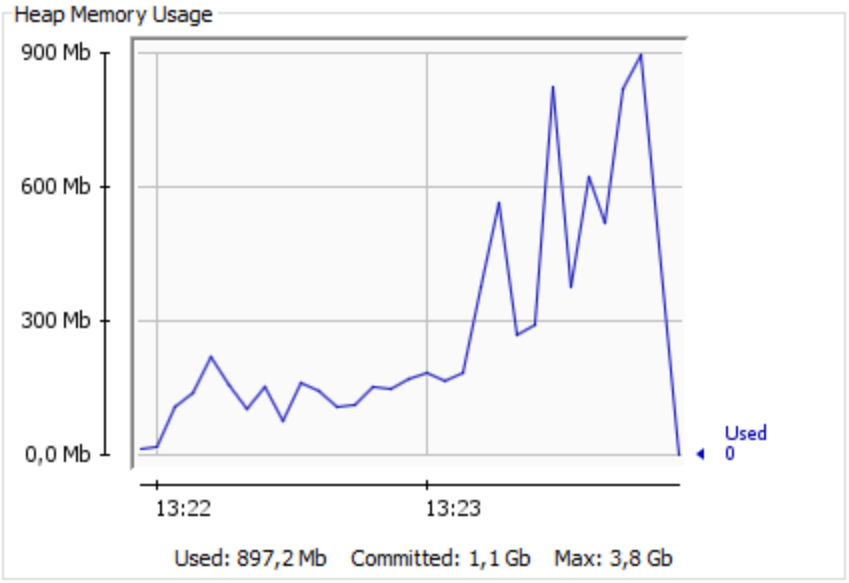

Figura 5.18: grafico utilizzo memoria RAM.

La memoria presenta un andamento crescente man mano che vengono lette le pagine delle aziende. I picchi sono dovuti alla lettura dei siti più grandi e sono seguiti da una discesa in quanto la **JVM** libera risorse quando sottoposta a carichi pesanti. La macchina virtuale Java, detta anche Java Virtual Machine o JVM, è il componente della piattaforma Java che esegue i programmi tradotti in bytecode dopo una prima compilazione. A default implementa una gestione che automaticamente è in grado di liberare le porzioni di memoria non più utilizzate.

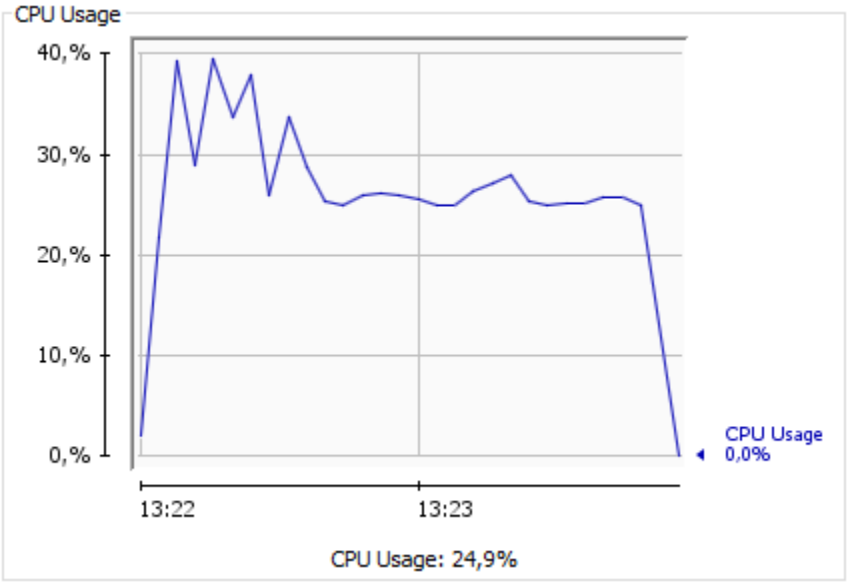

Figura 5.19: grafico utilizzo CPU.

La CPU dopo una fase iniziale di caricamento e inizializzazione del software si attesta mediamente su un utilizzo del 25%: questo è dovuto al fatto che la JVM utilizza solo uno dei quattro core a disposizione durante quasi tutta l'esecuzione.

Per quanto riguarda l'analisi delle singole fasi, vengono prese in esame la seconda (solo la parte legata alla lettura delle pagine) e la terza (ricerca dei KPI): infatti la prima viene eseguita solo una volta all'invocazione del programma ed essendo sempre uguale richiede un tempo fisso, mentre l'ultima richiede un tempo trascurabile in quanto si tratta di una breve scrittura su file e inoltre identica per tutte le aziende.
La prima fase richiede circa 70 millisecondi, mentre l'ultima 6-7 millisecondi per ciascuna azienda. Per quanto riguarda le altre due, sono state prese in considerazione solo le 12 aziende con più di cinquecento pagine, ordinate per numero decrescente, in quanto al di sotto di questa soglia le operazioni richiedono sempre meno di un secondo. Nel grafico di **Figura 5.20** vengono mostrati i tempi rilevati effettuando una media tra 3 esecuzioni.

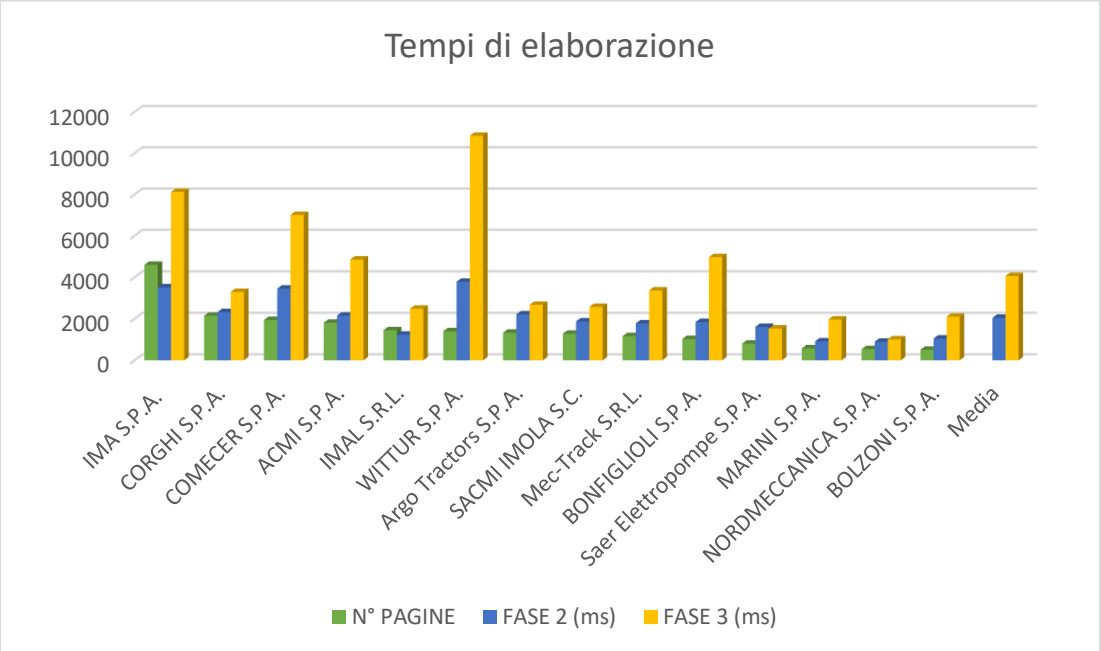

Figura 5.20: grafico dei tempi di elaborazione delle 12 aziende con >500 pagine.

I tempi tendono a diminuire al calare del numero di pagine, ma non c'è una relazione diretta tra i due. Per quanto riguarda la seconda fase, essa dipende molto dalla dimensione di ciascun

file da leggere. Invece la terza fase dipende da quanto testo è presente nel body, ovvero dalla lunghezza delle stringhe che vengono create in seguito alla lettura delle pagine.

L'unica eccezione riguarda l'azienda Wittur: i tempi infatti sono molto oltre la media, nonostante il numero di pagine non sia elevato. Analizzando le pagine scaricate, si è notato che esse presentano molto contenuto, che si traduce con più stringhe da analizzare durante la terza fase, allungando molto il tempo richiesto per l'elaborazione.

I valori misurati sono ottimi: il tempo medio per la seconda fase è 2 secondi, mentre per la terza servono mediamente 4 secondi.

Conclusioni

Analizzando i risultati è emerso che la ricerca è molto efficiente per quanto riguarda le certificazioni, sia ambientali che sociali, e comunque in generale per tutti i KPI formati da poche parole chiave. La situazione è invece più critica per gli Indicatori più lunghi, che spesso sono frasi di senso compiuto, i quali richiederebbero un'analisi diversa oppure un'ulteriore semplificazione e più parole addizionali. Probabilmente i risultati sarebbero migliorati modificando ulteriormente i KPI, cercando ancor più di adattarli, ma nonostante ciò il grado di precisione raggiunto è sicuramente soddisfacente.

Gli obiettivi prefissati sono stati raggiunti tutti. Il software realizzato è in grado di automatizzare il processo di download e ricerca dei KPI, ed è possibile eseguirlo anche prendendo in input molte aziende. La precisione ottenuta è ottima trattandosi di un primo tentativo di risolvere questo tipo di problema e anche le performance sono risultate molto buone.
Per quanto riguarda l'ultimo obiettivo rimasto, durante la progettazione si è sempre cercato di creare uno strumento che fosse indipendente da casi particolari e dall'ambito a cui è stato applicato: potenzialmente è in grado di essere utilizzato anche in contesti diversi da quello trattato, sia per quanto riguarda le aziende che gli indicatori da cercare.

Il software è ancora alle sue fasi iniziali e potrà essere migliorato sotto vari aspetti. Innanzitutto si può proseguire il lavoro di semplificazione dei KPI ed eventualmente tradurli in lingua inglese, per consentire una ricerca più ampia e completa. Molto interessante potrebbe essere l'estensione dell'analisi anche all'head delle pagine, in quanto possono contenere certe informazioni che però non vengono mostrate visitando il sito. Infine il software potrebbe considerare anche immagini, file PDF e quant'altro si trovi nei siti, per applicare l'analisi a tutti gli elementi presenti nei siti web.

Bibliografia e Sitografia

[Beckermann,1994] Beckermann W., *Sustainable Development: Is It a Useful Concept?*, 1994.

[Jabareen,2016] Jabareen Y.,*A New Conceptual Framework for Sustainable Development, Environment, Development and Sustainability*, 2008.

[UNEP,2009] UNEP, *Towards a Green Economy*, 2009.

[Mandrioli,2016] Mandrioli A., *Impresa sostenibile: raccolta e analisi dei dati per la realizzazione di un osservatorio sulle imprese in Emilia-Romagna. Un'analisi di benchmark nel settore dell'industria della fabbricazione di macchinari e attrezzature*, 2016.

http://www3.istat.it/strumenti/definizioni/ateco/ateco.html?versione=2007

https://www.gnu.org/software/wget/

https://jsoup.org/

http://opencsv.sourceforge.net/