Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

TARO: Infrastruttura per il Confronto di Testate Giornalistiche Internazionali

Relatore:
Prof.
ANGELO DI IORIO

Correlatore:
Prof.
GIOELE BARABUCCI

Presentata da:
GIUSEPPE CARRINO

Sessione I
Anno Accademico 2021/2022

A chi lotta
per informare e liberare

Indice

Elenco delle figure

Capitolo 1

Introduzione

I sistemi d'informazione hanno subito uno sviluppo notevole negli ultimi decenni: attualmente viviamo in un contesto popolato da centinaia di fonti, giornalistiche e non, attraverso le quali è possibile comprendere e analizzare il mondo che ci circonda. La grande quantità di fonti, tra testate, agenzie, siti web e altro ancora, è contemporaneamente sia uno strumento molto utile, per la possibilità di informarsi in maniera più ampia, sia un potenziale pericolo, poiché non è sempre facile capire su cosa fare affidamento. In questo contesto nasce l'idea di **TARO** (Tons of Articles Ready to Outline), un progetto che ha come scopo quello di confrontare articoli pubblicati su notiziari online per trarne conclusioni interessanti. Una prima parte del progetto, svolta durante un periodo di Erasmus+ Tirocinio presso l'NTNU, prende in considerazione i titoli degli articoli presi in esame, ma il progetto si è poi sviluppato analizzando anche i testi degli articoli stessi, cercando di aumentare l'accuratezza delle analisi svolte.

Ciò che risulta più significativo per il progetto è proprio capire quanto le fonti giornalistiche siano assimilabili, rispetto al numero di articoli pubblicati, ai canali di pubblicazione o alle tematiche delle notizie stesse: lo scopo è dunque quello di confrontare più testate online per capire quali argomenti siano trattati prevalentemente e quanto questi siano diversi tra le differenti fonti, paragonando, oltre ai temi, anche le modalità di divulgazione delle no-

tizie. Lo scopo di TARO è, in prima battuta, studiare alcune analisi possibili tra diverse fonti giornalistiche, in modo tale da porre le basi per delle metodologie di confronto, per poi definire **come** effettuare tali confronti effettuando analisi più specifiche, ad esempio comparando più algoritmi che risolvano lo stesso problema o aggregando le fonti in modi diversi. Queste valutazioni sono fatte implementando una pipeline estremamente modulabile che parta dalla raccolta delle informazioni dalle testate giornalistiche, fino a giungere alle analisi vere e proprie e alla loro visualizzazione, cercando di considerare tutte le variabili in gioco e di comprendere quali di esse portino a risultati soddisfacenti.

Lo sviluppo di questo progetto richiede una particolare attenzione sotto molteplici punti di vista: sarà necessario raccogliere articoli di giornale, allinearli per poterli confrontare, capire che tipo di confronti semantici siano rilevanti. La letteratura accademica non è priva di esperimenti di questo tipo, ma è necessario unire molteplici analisi per definire uno "stato dell'arte" relativo a questo argomento: è proprio la descrizione dello stato della tecnica il contenuto del *Capitolo 2*. Il *Capitolo 3 - Sviluppo del Progetto*, invece, servirà a spiegare quale sia la logica dietro la pipeline, esponendo criticità riscontrate e soluzioni proposte. Successivamente, nel *Capitolo 4 - Implementazione*, verrà esaminata l'implementazione puramente tecnica del progetto: le librerie utilizzate, le scelte implementative, gli input e gli output di ogni passo della pipeline. I risultati verranno mostrati e valutati nel *Capitolo 5 - Esperimento*, che descriverà quanto rilevato dalle analisi effettuate tra le due agenzie giornalistiche italiane AGI e ANSA utilizzando la pipeline sviluppata. Il *Capitolo 6* elencherà i potenziali punti critici del progetto, mentre il *Capitolo 7* servirà a trarre le conclusioni, oltre che a descrivere i miglioramenti possibili e i possibili utilizzi futuri di TARO.

Capitolo 2

Stato dell'arte

Per la comprensione degli obiettivi di TARO e del tipo di analisi descritte in questa tesi è utile discutere vari aspetti dello stato della tecnica, in merito alla raccolta e all'analisi di articoli di giornale pubblicati online. Più nel dettaglio, è utile approfondire quali siano gli algoritmi maggiormente utilizzati per effettuare confronti tra testi: in particolare, data la necessità di valutare quali testate trattino gli stessi argomenti, è importante comparare diversi algoritmi di similarità semantica, al fine di stabilire quali siano le metodologie più adatte per gli scopi di TARO.

Verrà dunque chiarito lo stato dell'arte per due particolari aspetti: l'analisi di notiziari online e gli algoritmi di similarità semantica.

2.1 Analisi di notiziari

Particolarmente rilevante per il tipo di analisi di articoli di giornale è lo studio di articoli spagnoli e italiani sulla pandemia di COVID-19 di Tejedor et al. [12]: in tale studio sono state definite diverse metodologie di organizzazione dei documenti sotto forma di Cluster, tecnica sfruttata anche per TARO. Inoltre, la ricerca conclude anche evidenziando la politicizzazione degli articoli presi in esame, dimostrando quanto le analisi su testate giornalistiche possano condurre a valutazioni interessanti anche sotto l'aspetto

socio-culturale di diversi Stati. Un'ulteriore ricerca che prende in considerazione articoli sul COVID-19 è quella di Zafri et al. [17], che invece sfrutta la libreria NLTK e il modello LDA[11] per dividere i vari articoli in categorie: tale tecnica non è stata sfruttata per questo progetto, ma risulta estremamente interessante per eventuali miglioramenti futuri. Riguardo alla NamedEntityRecognition, uno studio effettuato da Shelar et al. [14] ha evidenziato come SpaCy fosse la libreria più indicata per effettuare tale analisi, rispetto a TensorFlow e OpenNLP, di conseguenza è stata proprio questa la libreria scelta per la NER in questo progetto.

2.2 Similarità tra contenuti di testi

Uno degli algoritmi più utilizzati attualmente per calcolare la vicinanza semantica tra testi è l'algoritmo del *Coseno*. Questo algoritmo si basa sul calcolo di un vettore di occorrenze dei termini presenti nei due testi confrontati: il funzionamento dell'algoritmo è spiegato da Lahitani et al. [6], esponendo anche il funzionamento del peso TF-IDF, utilizzato in questo progetto. L'utilizzo di questo algoritmo è stato testato anche per il lavoro di Berlanga-Llavori [10], in particolare proprio per il confronto di articoli di giornali, al fine di identificare eventi-chiave. Per quanto riguarda invece l'utilizzo del PosTagging (in seguito usato per l'algoritmo **TermMatching**) è stata fondamentale l'intuizione di Sandhya e Govardhan [13], corredata anche dall'utilizzo di WordNet per l'allineamento dei termini: il dizionario non è stato implementato in questo progetto, ma sarebbe certamente interessante migliorare TARO proprio grazie a WordNet e le sue potenzialità. Lo stesso articolo, inoltre, mostra come anche Jaccard e Pearson possano essere utilizzati in futuro come algoritmi di similarità, oltre al già citato algoritmo del Coseno. Il PosTagging è anche indicato come particolarmente utile da un articolo di Liu e Curran [7] e uno di Yalcine et al. [16]: la prima ricerca mette anche in luce l'importanza della lemmificazione come strumento di analisi semantica, mentre la seconda sfrutta in particolare il word-embedding.

Capitolo 3

Sviluppo del Progetto

L'obiettivo di questo capitolo è quello di fornire una panoramica sul progetto svolto. La pipeline realizzata mira a creare un infrastruttura utilizzabile anche in futuro per l'analisi e il confronto di notizie pubblicate sul Web, partendo dall'estrazione delle stesse, definendo il tipo di fonti e il metodo di raccolta dei dati, passando per significative analisi semantiche, decidendo quali siano rilevanti per lo sviluppo del progetto, e giungendo a dei valori che descrivano la copertura di determinate tematiche da parte di diverse fonti giornalistiche, applicando due algoritmi di confronto tra testi.

3.1 Contesto e Problema

La finalità di questo progetto è confrontare tra loro diverse notizie per comprendere di cosa parlano, e in che modo lo fanno, diverse testate, per poter definire, ad esempio, percentuali di copertura di argomenti (quante notizie hanno trattato entrambe le fonti in oggetto) o di esclusività (quante notizie sono state riportate solo da una delle fonti). In questo contesto, quindi, *accoppiare notizie* significa confrontarle e trarre come conclusione il fatto che esse stiano trattando lo stesso argomento. Prese dunque due testate si vuole poter collegare come simili due o più notizie tra loro, come mostrato in *Figura 3.1*.

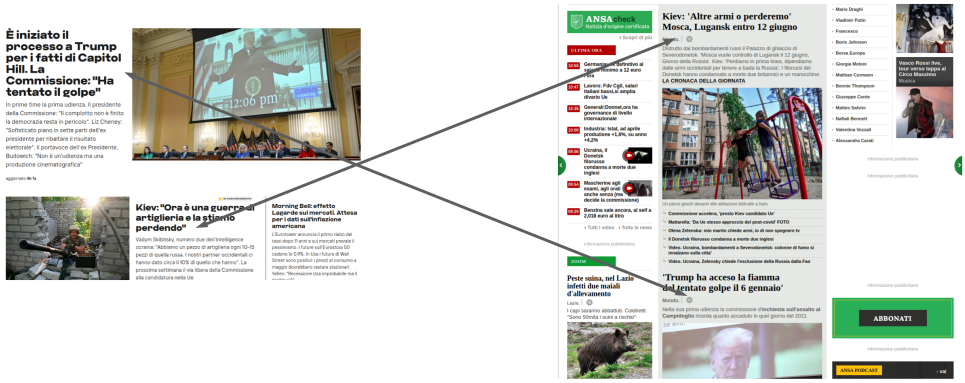


Figura 3.1: Visualizzazione grafica di accoppiamenti tra notizie simili

Le fonti giornalistiche online risultano molto distanziate tra loro, sia in merito alla forma espressiva che alla gestione dei flussi di notizie: l'obiettivo è dunque scegliere esattamente cosa siano le "notizie" che verranno confrontate, cioè quali elementi di un eventuale sito web giornalistico siano le unità da confrontare successivamente e come sono caratterizzati in merito alla frequenza di pubblicazione o alla lingua, ad esempio.

Da questa distanza deriva la distinzione tra testate a **Edizione** e testate a **Flusso**:

- Testate a **Edizione**: notizie fornite periodicamente, come ad esempio telegiornali o radiogiornali. Nel caso in esame, si è trattato di notizie raccolte a cadenza quotidiana, al fine di comparare articoli relativi allo stesso giorno. Esempi di testate ad edizione sono il "il TG1 delle 20:00" o "l'edizione a stampa di Repubblica";

- Testate a **Flusso**: notizie fornite a flusso continuo, come ad esempio giornali online, ottenendo quindi degli "snapshot" a cadenza regolare che fornissero un quadro del notiziario in diversi momenti della giornata. Esempi di testate ad edizione sono "l'edizione online di Repubblica" o "il flusso RSS di ANSA".

In particolare, in questa tesi, le notizie reperite da testate a Flusso verranno analizzate più a fondo, portando con sé, tuttavia, problematiche relative ai canali di pubblicazione delle stesse: come punti di partenza, sono stati ritenuti interessanti i Feed RSS e le Homepage dei giornali online. I primi riportano le notizie in formato *XML* e sono suddivisi per canale in base alle tematiche, ad esempio un canale per la politica o uno per gli esteri. Le Homepage, invece, variano in base alle sezioni del sito, secondo la stessa logica dei diversi Feed: per la prima parte del progetto, si è scelto di analizzare i canali relativi agli *Esteri*, al fine di massimizzare la probabilità di rilevare notizie simili.

Le fonti scelte, a prescindere dai canali di pubblicazione, sono scritte in lingue diverse: sono state selezionate proprio per effettuare delle analisi comparative più interessanti. Tuttavia, per poter effettuare analisi di similarità sarà necessario allineare linguisticamente le notizie ottenute, intuitivamente, traducendole tutte in una sola lingua per poi poter sfruttare algoritmi di similarità semantica.

3.2 Panoramica della Soluzione

Di seguito sono visualizzati i passi logici del progetto.

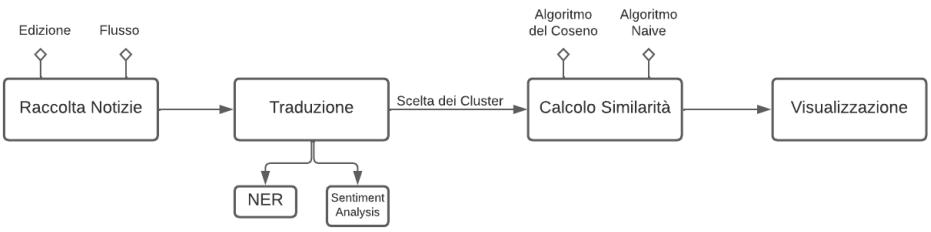


Figura 3.2: Passi logici del progetto

I passi sono quindi:

1. Raccolta delle notizie, ovvero lo scraping degli articoli dalle Homepage o dai Feed RSS delle fonti scelte

2. Traduzione, data la necessità di allineare linguisticamente i notiziari di Paesi diversi per poterli comparare.

3. Calcolo della similarità, che consiste nel confronto di due notizie (prese da insiemi ben delineati) al fine di stabilirne la vicinanza nelle tematiche

4. Visualizzazione grafica dei risultati ottenuti

3.3 Fonti Eterogenee

Il primo passo per lo sviluppo della pipeline è la scelta delle fonti. L'idea è quella di selezionare testate disomogenee per due motivi principali: effettuare analisi più interessanti e capire punti di forza e limiti della pipeline stessa.

Sono stati quindi scelte fonti, sia a edizione che a flusso, che coprissero diverse lingue e diverse linee editoriali. In particolare, le testate selezionate sono riportate nella tabella seguente.

Edizione		**Flusso**	
RTS	*www.rts.ch/*	*Spiegel*	*ww.spiegel.de/*
*France*24	*www.france*24*.com/en/*	*France*24	*www.france*24*.com/fr/*
Tagesschau	*www.tagesschau.de/*	*BBC*	*www.bbc.co.uk/*
ZDF	*www.zdf.de/*	*CNN*	*edition.cnn.com/*
PBS	*www.pbs.org/*	*ABC*	*abc.com/*
*GR*1	*www.raiplaysound.it/*	*ilPost*	*www.ilpost.it/*

Tabella 3.1: Testate scelte come fonti

Risultano quindi quattro testate in lingua inglese (France24 a Edizione, PBS, BBC, CNN), tre in tedesco (Tagesschau, ZDF, Spiegel), due in italiano (GR1, ilPost) e francese (France24 a Flusso, RTS) e una in spagnolo (ABC).

Inoltre, si può facilmente notare anche una certa differenza editoriale tra le varie testate, avendo deciso di comparare, ad esempio, testate come ilPost, che produce un gran numero di editoriali, e testate come CNN, più assimilabile a un feed continuo di notizie contestuali.

In una prima fase del progetto, sviluppatasi durante un tirocinio Erasmus presso il NTNU, è stata data particolare enfasi alle notizie a edizione, ritenendole più significative e interessanti: inoltre, il confronto di testate a flusso porta con sé la difficoltà di scegliere determinati frame temporali di confronto (vedasi *sezione 3.3.1*).

In seguito si è deciso invece di lavorare maggiormente proprio su quest'ultima tipologia di fonti, sia per l'importanza di lavorare su variabili nuove, come, appunto, la scelta di frame temporali, sia per la possibilità di ottenere anche un vero proprio corpo della notizia, analizzabile in maniera più efficace.

I canali dai quali le notizie sono state reperite sono molteplici: le notizie delle fonti a edizione sono ottenute attraverso il Web-Scraping, con oggetto gli archivi dei vari telegiornali e i relativi titoli di testa, ognuno dei quali rappresentante una notizia. Le notizie delle fonti a flusso sono invece ottenute in due diversi modi:

- **Feed RSS**: utilizzo dei canali RSS forniti dai giornali stessi per ottenere le notizie e i relativi metadati in un dato momento della giornata. La scelta degli specifici canali e la loro omogeneità costituirà parte fondamentale della *sezione 3.5*;
- **Scraping delle Homepage**: seguendo lo stesso principio dello scraping per le testate a edizione, vengono analizzate le pagine principali dei giornali online e ne vengono estratte le notizie pubblicate, ottenendo uno "snapshot" delle notizie in un dato istante. Questa tecnica è

stata utilizzata solo per ilPost e Televideo, a causa dell'assenza di Feed RSS.

Le testate a edizione hanno seguito una pipeline diversa da quella a flusso, a causa delle differenze strutturali delle due tipologie: le notizie derivate da fonti a edizione, per la maggior parte, non hanno un contenuto, ma solo un titolo, essendo reperite dalle intestazioni di telegiornali.

Verrà quindi chiarito quando le due pipeline seguiranno linee differenti e ne verranno messe in luce le differenze.

Le seguenti sezioni mirano ad analizzare due delle principali scelte implementative effettuate per ottenere i risultati desiderati: queste scelte risulteranno essere anche potenziali limiti dell'approccio (vedasi *Capitolo 6 - Threats to Validity*).

3.3.1 Coperture Eterogenee: Scelta Frame Temporali

I **frame temporali** sono definiti come un lasso di tempo significativo per il raggruppamento di notizie diverse, e, come esposto in precedenza, una criticità dei confronti delle fonti a flusso risulta essere proprio la definizione dei frame temporali da comparare. Come mostrato nella *Figura 3.3*, le notizie vengono reperite dalle fonti attraverso degli *"snapshot"*, ovvero si immagazzinano gli articoli presenti in un determinato momento sul canale della fonte in oggetto (che si tratti di Feed RSS o Homepage): gli *snapshot* sono rappresentati dai blocchi interni, identificati dalla data e l'ora di reperimento, e relativi a una specifica fonte. I Frame Temporali, invece, sono degli insiemi di snapshot, raggruppati in base all'ora di reperimento: nell'esempio, sono presenti frame di un'ora (dalle 17 alle 17:59) e sono proprio tali insiemi ad essere comparati, ovvero notizie ottenute da uno snapshot delle 17 della Fonte A verranno confrontate, ad esempio, anche con articoli raccolti alle 17:30 dalla Fonte B.

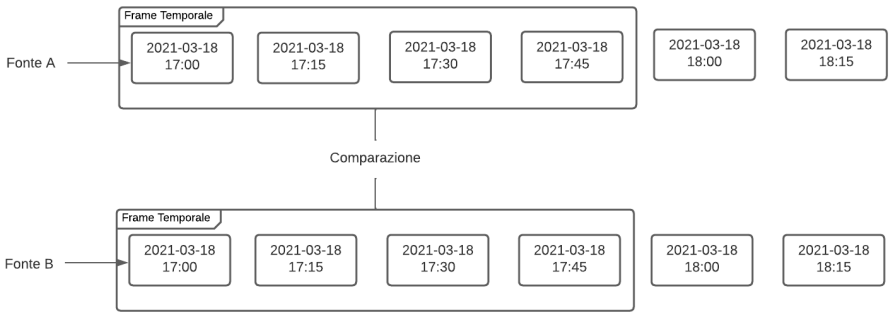


Figura 3.3: Visualizzazione grafica di snapshot e frame temporali

La definizione di *snapshot* riportata sopra è necessariamente derivata dalla natura delle testate a flusso: queste testate pubblicano notizie in modo disomogeneo e continuativamente durante tutta la giornata. Per questo, gli *snapshot* vengono utilizzati per "catturare" tale flusso continuo, rappresentando la situazione, in diversi momenti, del canale di pubblicazione del notiziario.

Si potrebbe pensare di considerare allineati gli *snapshot* della stessa ora, ad esempio, in *Figura 3.3*, confrontare le notizie delle 17:00 di *Fonte A* con quelle delle 17:00 di *Fonte B*, ragionando in modo pressochè analogo a quanto deciso per le testate a edizione (confrontare edizioni del giorno tra loro). Tuttavia, il confronto tra *snapshot* così ravvicinati risulta intuitivamente poco interessante: sarebbe improbabile trovare notizie simili su testate diverse nello stesso istante, dato che, per quanto riguarda le notizie flusso, ci si aspetterebbero diversi orari di pubblicazione della medesima notizia, dipendentemente da

- Rilevanza della notizia,
- Linea editoriale,
- Distanza geografica della testata dall'avvenimento, ecc.

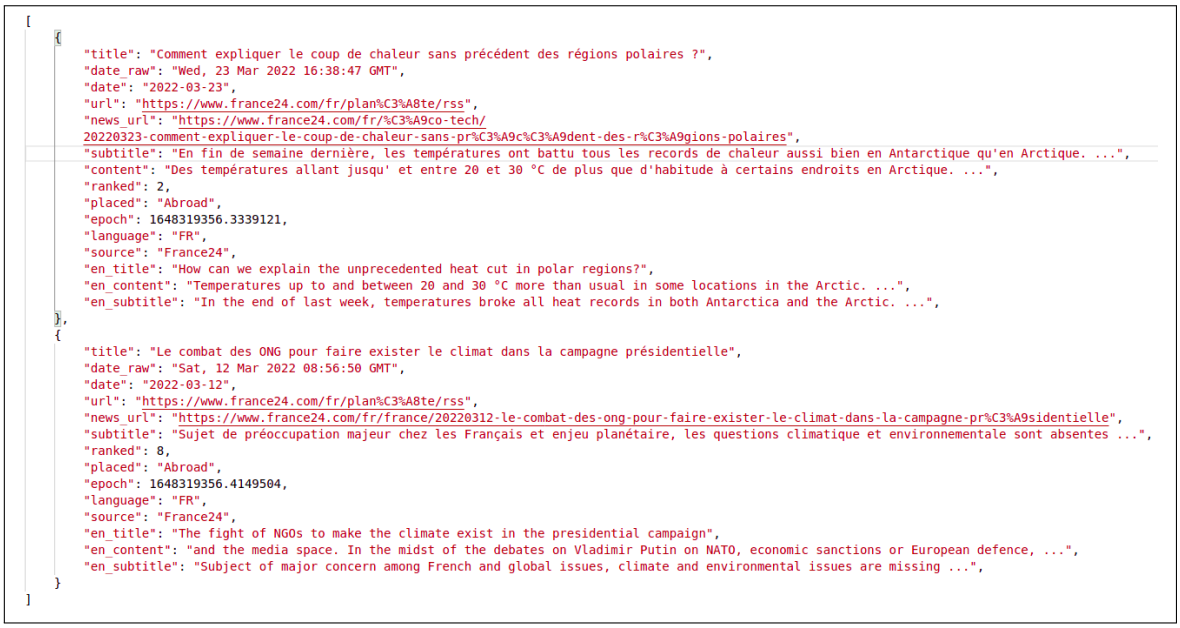
Di conseguenza, è risultato necessario definire dei lassi tempo all'interno dei quali raggruppare molteplici "snapshot" delle diverse fonti, comparando poi gli interi gruppi tra loro.

Quanto sopra spiega la necessità di un *limite inferiore* relativo alla grandezza dei frame, ovvero un minimo lasso di tempo da considerare per la scelta dei frame, per evitare di perdere confronti significativi. Ciononostante, è anche importante definire un *limite superiore* per evitare sia tempi di computazione troppo lunghi, che percentuali di similarità tra notiziari troppo basse, dovute alla presenza massiccia di notizie specifiche per una o l'altra testata.

Per questo progetto sono stati quindi scelti due valori significativi:

- **15 minuti** come cadenza dei vari "snapshot";
- **1 ora** come frame temporale di confronto.

Ciò implica il raggruppamento di quattro istantanee ciascuno per ogni confronto. In *Figura 3.4* è riportato un esempio di snapshot ottenuto da una fonte a flusso: verrà chiarito nel *Capitolo 4* come tale raccolta è stata implementata.



```
[
    {
        "title": "Comment expliquer le coup de chaleur sans précédent des régions polaires ?",
        "date_raw": "Wed, 23 Mar 2022 16:38:47 GMT",
        "date": "2022-03-23",
        "url": "https://www.france24.com/fr/plan%C3%A8te/rss",
        "news_url": "https://www.france24.com/fr/%C3%A9co-tech/
        20220323-comment-expliquer-le-coup-de-chaleur-sans-pr%C3%A9c%C3%A9dent-des-r%C3%A9gions-polaires",
        "subtitle": "En fin de semaine dernière, les températures ont battu tous les records de chaleur aussi bien en Antarctique qu'en Arctique. ...",
        "content": "Des températures allant jusqu' et entre 20 et 30 °C de plus que d'habitude à certains endroits en Arctique. ...",
        "ranked": 2,
        "placed": "Abroad",
        "epoch": 1648319356.3339121,
        "language": "FR",
        "source": "France24",
        "en_title": "How can we explain the unprecedented heat cut in polar regions?",
        "en_content": "Temperatures up to and between 20 and 30 °C more than usual in some locations in the Arctic. ...",
        "en_subtitle": "In the end of last week, temperatures broke all heat records in both Antarctica and the Arctic. ...",
    },
    {
        "title": "Le combat des ONG pour faire exister le climat dans la campagne présidentielle",
        "date_raw": "Sat, 12 Mar 2022 08:56:50 GMT",
        "date": "2022-03-12",
        "url": "https://www.france24.com/fr/plan%C3%A8te/rss",
        "news_url": "https://www.france24.com/fr/france/20220312-le-combat-des-ong-pour-faire-exister-le-climat-dans-la-campagne-pr%C3%A9sidentielle",
        "subtitle": "Sujet de préoccupation majeur chez les Français et enjeu planétaire, les questions climatique et environnementale sont absentes ...",
        "ranked": 8,
        "placed": "Abroad",
        "epoch": 1648319356.4149504,
        "language": "FR",
        "source": "France24",
        "en_title": "The fight of NGOs to make the climate exist in the presidential campaign",
        "en_content": "and the media space. In the midst of the debates on Vladimir Putin on NATO, economic sanctions or European defence, ...",
        "en_subtitle": "Subject of major concern among French and global issues, climate and environmental issues are missing ...",
    }
]
```

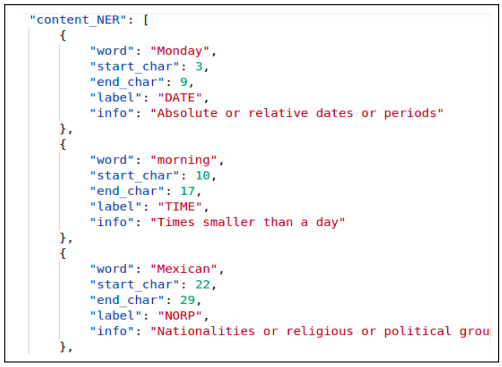

Figura 3.4: Estratto di snapshot di notizie da notiziari a flusso

Verrà approfondita in seguito la necessità di modificare dinamicamente l'ampiezza dei frame, per poter effettuare analisi più significative.

3.3.2 Eterogeneità Qualitativa: Confronto tra lingue diverse

Un'ulteriore criticità relativa all'allineamento delle notizie è derivato dalle diverse lingue dei notiziari.

Le notizie reperite da fonti a flusso, in quanto costituite anche da un campo relativo al contenuto, di lunghezza spesso significativa, hanno reso importante un'analisi costi-benefici della traduzione stessa: sono state quindi eseguite alcune analisi semantiche, come *NER* e *Sentiment Analysis*, al fine di confrontare i risultati ottenuti da notizie tradotte in lingua inglese con quelli ricavati da librerie che lavorano su lingue diverse. Due NER di questo tipo sono messe a confronto in *Figura 3.5*.

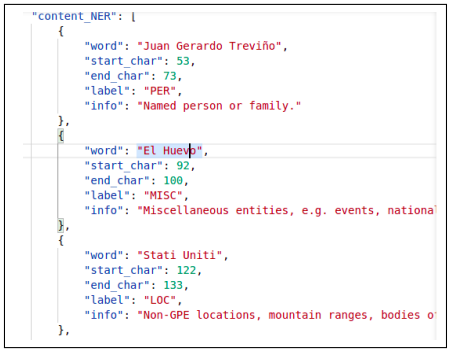


```
"content_NER": [
    {
        "word": "Monday",
        "start_char": 3,
        "end_char": 9,
        "label": "DATE",
        "info": "Absolute or relative dates or periods"
    },
    {
        "word": "morning",
        "start_char": 10,
        "end_char": 17,
        "label": "TIME",
        "info": "Times smaller than a day"
    },
    {
        "word": "Mexican",
        "start_char": 22,
        "end_char": 29,
        "label": "NORP",
        "info": "Nationalities or religious or political grou
    },
```


(a) Testo tradotto (1)

```
{
    "word": "Juan Gerardo Treviño",
    "start_char": 51,
    "end_char": 71,
    "label": "PERSON",
    "info": "People, including fictional"
},
{
    "word": "El Huevo",
    "start_char": 83,
    "end_char": 91,
    "label": "WORK_OF_ART",
    "info": "Titles of books, songs, etc."
},
{
    "word": "the United States",
    "start_char": 107,
    "end_char": 124,
    "label": "GPE",
    "info": "Countries, cities, states"
},
```


(b) Testo tradotto (2)



```
"content_NER": [
    {
        "word": "Juan Gerardo Treviño",
        "start_char": 53,
        "end_char": 73,
        "label": "PER",
        "info": "Named person or family."
    },
    {
        "word": "El Huevo",
        "start_char": 92,
        "end_char": 100,
        "label": "MISC",
        "info": "Miscellaneous entities, e.g. events, nationa
    },
    {
        "word": "Stati Uniti",
        "start_char": 122,
        "end_char": 133,
        "label": "LOC",
        "info": "Non-GPE locations, mountain ranges, bodies o
    },
```


(c) Testo in lingua originale

Figura 3.5: Esempio di confronto tra NER in lingua originale e NER in seguito a traduzione

Come visibile in figura, data la maggiore quantità di entità riconosciute dal testo tradotto, questo confronto ha portato a conclusioni, per quanto contestuali agli obiettivi preposti dal progetto, favorevoli alla traduzione: le analisi ottenute con algoritmi specifici per le varie lingue risultavano poco accurate e ciò, in aggiunta alla minore efficienza di tali algoritmi, ha portato alla decisione di accantonare per il momento questo filone di ricerca, in favore di un allineamento delle notizie in lingua inglese.

La poca esattezza sopracitata si discosta anche dai risultati ottenuti da Pastor et al.[2] riguardo l'accuratezza di analisi effettuate su testi tradotti: tale discrepanza è tuttavia riconducibile sia alla natura dei testi da loro ana-

lizzati (trattasi di testi medico-scientifici), sia alla scarsità di strumenti volti all'analisi semantica nelle lingue qui prese in oggetto.

Come esempio di dataset per la NER in lingua originale, è stato utilizzato *KIND*[9], mentre per quanto riguarda la sentiment analysis in italiano è stata utilizzata la libreria *feel-it*[1].

Le informazioni ottenute tramite *Sentiment Analysis* saranno comunque utilizzate per fare confronti (vedasi *sezione 3.6*).

Per quanto riguarda la pipeline seguita dalle testate a edizione, data anche la piccola mole di dati da comparare (poche notizie rispetto ai flussi e vengono comparati solo titoli), è stato ritenuto conveniente tradurre quanto rilevante in inglese, utilizzando le API di GoogleTranslate[2].

3.4 Calcolo della similarità

Una volta allineate le notizie, sia in termini qualitativi (**linguistici**) che quantitivi (**temporali**), è necessario definire le modalità e le finalità dei confronti.

L'algoritmo per il calcolo della similarità si basa sulla ricerca delle parole chiave delle notizie per calcolarne un certo grado di similarità semantica, di modo da poter definire notizie che trattino dello stesso argomento.

Il risultato è un insieme di confronti tra tutte le coppie possibili di notizie dei due frame presi in esame, con un determinato numero di notizie considerate tra loro sovrapponabili in merito agli argomenti trattati: gli accoppiamenti non sono biunivoci, questo per evitare di escludere notizie che da un notiziario potrebbero essere trattate da più articoli rispetto all'altro.

[1] https://github.com/MilaNLProc/feel-it

[2] https://cloud.google.com/translate/docs/reference/rest

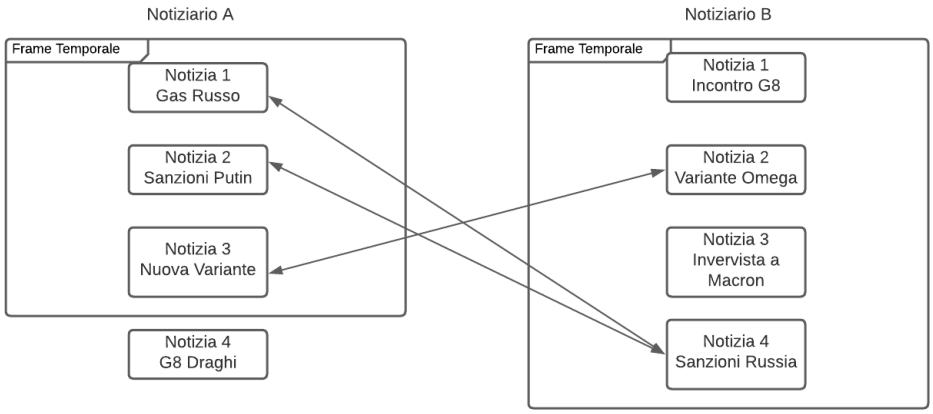


Figura 3.6: Schema logico del confronto tra due notiziari: in questo caso, la cardinalità del primo frame è pari a 3, mentre quella del secondo è pari a 4. Le frecce indicano notizie che trattano lo stesso argomento

Per quanto riguarda la pipeline delle notizie a edizioni, a causa della quantità di dati più ristretta, si è ritenuto interessante fare un'analisi particolarmente specifica, ovvero cercare notiziari che riportassero esattamente la stessa notizia nello stesso giorno, cercando una certa similarità semantica nel titolo delle stesse. A tal fine e' stata utilizzata una libreria di NLP per fare pos-tagging del testo da comparare (vedasi *sezioni 3.4.1-3.4.2*).

	Bern: the session of women rejected for two days to the Federal Parliament		Rome: meeting between Joe Biden and Emmanuel Macron at the G20		A semiconductor penetration turns the automotive industry in slow motion	
Biden tells Macron US was clumsy over Australian submarine deal	session women days Federal Parliament	Biden Macron US clumsy submarine deal	Rome meeting Joe Biden Emmanuel Macron G20	Biden Macron US clumsy submarine deal	semiconductor penetration industry motion	Biden Macron US clumsy submarine deal
Women and the climate crisis: COP26 talks seen as last chance	session women days Federal Parliament	Women climate crisis COP26 talks chance	Rome meeting Joe Biden Emmanuel Macron G20	Women climate crisis COP26 talks chance	semiconductor penetration industry motion	Women climate crisis COP26 talks chance

Figura 3.7: Esempio di confronti fatti per testate a Edizione (Algoritmo TermMatching).Le due colonne di termini per ciascuna cella indicano i rispettivi "concetti" di ognuna delle notizie comparate, quelli in verde sono i concetti uguali tra i due titoli, che implicano una sovrapponibilità

Le testate a flusso sono invece comparate sia utilizzando pressoché lo stesso algoritmo utilizzato per le testate a edizione (in seguito definito *TermMatching*), sia l'algoritmo del Coseno, che risulta efficiente su testi particolarmente lunghi, caratteristica che lo rende poco utile per le notizie reperite da fonti a edizione.

3.4.1 Estrazione Informazioni Lessicali

Il primo passo per l'esecuzione dell'algoritmo *TermMatching* è l'analisi grammaticale del testo, finalizzata alla classificazione di tutte le parole che lo compongono. Di seguito sono riportati alcuni dei "tag" apposti agli elementi del testo dal Pos-Tagger.

Tag	Descrizione	Esempio
NOUN	Nome comune	Gatto, tavolo, camicia
PROPN	Nome proprio	Elisa, Lorenzo, Ludovica
VERB	Verbo	Scrive, corro, mangi
ADJ	Aggettivo	Alto, largo, rotto

Tabella 3.2: Esempi di tag attribuiti dal Pos-Tagger

3.4.2 Algoritmo TermMatching con Pos-Tag

Una volta ottenuta una classificazione lessicale grazie al Pos-Tagging, risulta necessario capire **cosa** comparare e **come** farlo. L'idea dietro l'algoritmo *TermMatching* è individuare come parti significative del testo i nomi comuni (*NOUN*) e i nomi propri (*PROPN*): questi saranno l'oggetto del confronto e verranno da ora chiamati **concetti** per semplicità.

Si è deciso di escludere parti del discorso come i verbi (*VERB*) a causa dell'alta probabilità di traduzioni poco contestualizzate, che avrebbero più facilmente compromesso i risultati finali dei confronti, a differenza di nomi comuni e propri che tendono a subire meno la *perdita nella traduzione*.

La comparazione effettuata da **TermMatching** è un vero proprio confronto tra i *concetti* delle due notizie, con lo scopo di identificare la quantità di concetti uguali tra esse. È stata quindi decisa una percentuale minima di concetti uguali tra due notizie affinchè queste venissero ritenute *simili*: il numero viene chiaramente normalizzato rispetto alla lunghezza degli articoli per evitare

- sia, che articoli molto prolissi vengano accoppiati troppo facilmente (**falsi positivi**),
- sia, che articoli brevi non riscontrino alcuna somiglianza con altri (**falsi negativi**).

Uno schema logico esplicativo dell'algoritmo in questione è visibile in *Figura 3.8*.

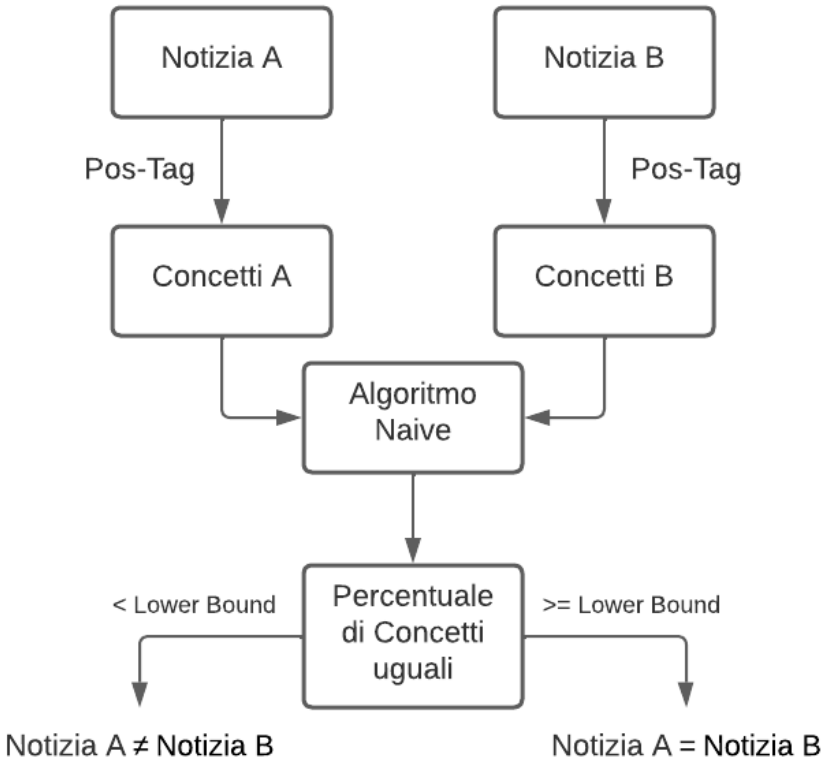


Figura 3.8: Schema logico dell'algoritmo TermMatching

Tale soglia, per questo progetto, è stata fissata al 10% della media dei concetti delle due notizie: il valore non è particolarmente elevato a causa della scelta delle parti di testo significative, data la grande presenza di nomi comuni irrilevanti nel testo. Inoltre, il valore è normalizzato rispetto alla media dei concetti delle due notizie proprio per i motivi sopracitati (la media risulta consistente data l'omogeneità degli ordini di grandezza delle notizie).

Grazie a questo algoritmo i confronti risultano abbastanza efficienti e sono rari i falsi positivi: va però considerato il consistente costo computazionale

in termini di tempo di esecuzione, dovuto all'estrazione delle informazioni (vedasi *sezione 3.4.1*).

Come esposto nelle precedenti sezioni, i confronti sono state effettuati tra snapshot in frame temporali di un'ora ciascuno: ciò è stato iterato per 24 ore, al fine di monitorare la variazione di percentuale di notizie simili presenti tra diverse testate nell'arco di una giornata.

Nel grafico seguente (*Figura 3.9*) è possibile vedere un esempio della variazione di percentuale sopracitata.

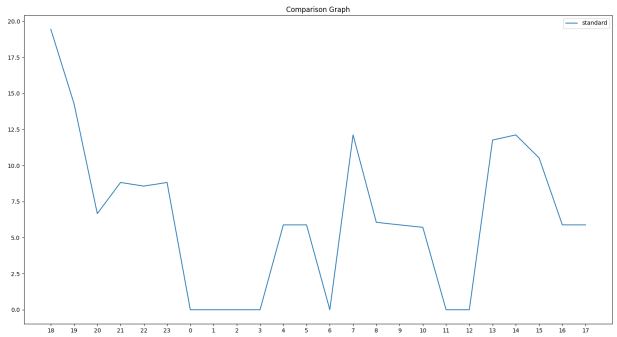


Figura 3.9: Esempio di variazione percentuale nell'arco di una giornata di notizie simili (BBC - CNN 2022/04/21, algoritmo TermMatching)

Si possono notare sia picchi massimi di percentuale relativamente soddisfacenti (20% alle 18), sia minimi molto bassi (0% tra le 11 e le 12).

Per certificare questi risultati si è implementato anche un ulteriore algoritmo di similarità: quello del Coseno.

3.4.3 Algoritmo del Coseno

L'algoritmo del coseno si basa sul calcolare la percentuali di termini uguali tra entrambi i documenti, analizzando l'intera stringa (senza scegliere parti del discorso particolari, come per l'algoritmo *TermMatching*), ma dando me-

no peso ai termini troppo frequenti, al fine di evitare falsi positivi dovuti, ad esempio, a **stop-words**.

A tal fine, vengono prima eseguite una **tokenization** e **lemmization**, per allineare tutti i termini, ad esempio coniugando tutti i verbi all'infinito e declinando i nomi al singolare. Vengono quindi costruiti degli array delle frequenze ed applicata la seguente formula:

$$S = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figura 3.10: Equazione del Coseno: con A_i e B_i si intendono il numero di occorrenze del termine i-esimo rispettivamente nella stringa A e nella stringa B, S è un numero reale che varia da 0 a 1 indice della similarità

L'algoritmo del coseno è particolarmente indicato per il calcolo delle similartà tra testi [15], confrontandolo con l'algoritmo di Dice e l'algoritmo di Jaccard.

Le stesse analisi definite in precedenza sono state effettuate anche con l'algoritmo in questione e di seguito è riportato il grafico di similarità (*Figura 3.11*).

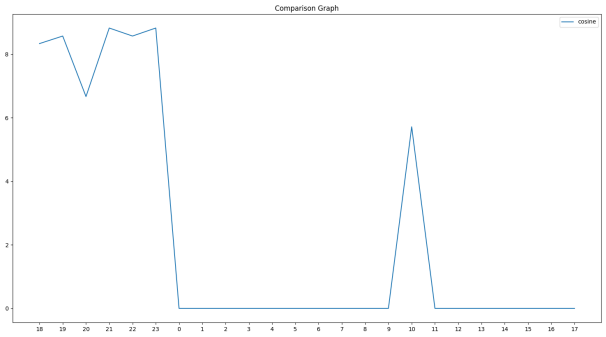


Figura 3.11: Esempio di variazione percentuale di notizie simili (BBC - CNN 2022/04/21, algoritmo del Coseno)

Anche in questo caso vi sono alcuni picchi, presumibilmente all'occorrenza di una notizia di particolare rilevanza, ma sono frequenti i frame temporali con una percentuale di similarità pari allo 0%. Di seguito, in *Figura 3.12*, sono riportati gli stessi grafici comparati.

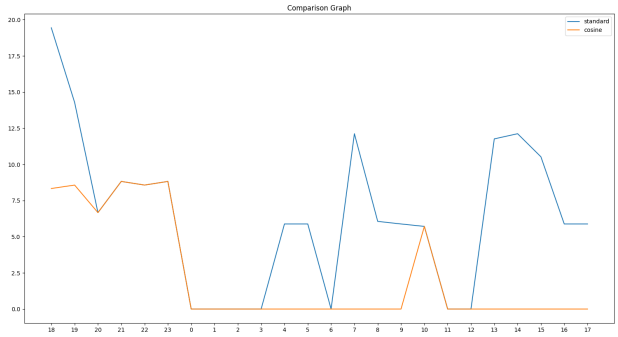


Figura 3.12: Esempio di variazione percentuale di notizie simili (BBC - CNN 2022/04/21, algoritmi comparati)

Si può notare come tendenzialmente i due algoritmi seguano le stesse oscillazioni, ma l'algoritmo *TermMatching* presenti un maggior numero di picchi di similarità e, in generale, abbia delle percentuali di similarità media

maggiori: ciò è dovuto alla naturale maggiore precisione dell'algoritmo del coseno, che tende ad accoppiare solo notizie fortemente correlate.

Anche per quanto riguarda quest'ultimo algoritmo è stato necessario definire un **limite inferiore** alla percentuale di similarità che definisse una sovrapposizione (quasi) completa tra gli oggetti del confronto: in questo caso si è scelta una percentuale di similarità calcolata dall'algoritmo dell'80%.

La percentuale può risultare alta, ma ciò è dovuto alla non-linearità dell'algoritmo del coseno: notizie considerabili sovrapponibili intuitivamente, avranno percentuali di similarità anche del 90%, tuttavia tra il 40% e il 70% di similarità si posizionano coppie anche molto distanti semanticamente. Si possono visionare ulteriori esempi di grafici relativi a confronti tra Spiegel e France24, sia in merito alle coperture di notizie che alla Sentiment Analysis nell'*Appendice A* (*Figura A.1*, *Figura A.2*, *Figura A.3*).

3.5 Cluster Temporali e Feed Cluster

I risultati ottenuti, tuttavia, non risultavano abbastanza convincenti: ciò che appare poco probabile è la percentuale di similarità, in valore assoluto, per ciascun frame temporale. Intuitivamente, infatti, ci si aspetterebbe tra giornali nella stessa lingua e con linee editoriali simili (come sopra *CNN* e *BBC*) una percentuale di notizie che trattano lo stesso argomento molto alta, ad un orario stabilito.

Sono stati quindi controllati diversi file di confronto al fine di verificare la presenza di falsi negativi (o positivi), che tuttavia, dopo un'analisi manuale, non sono stati riscontrati in maniera significativa.

Questa intuizione ha portato a voler definire un **Golden Standard** che esemplificasse i confronti e desse validità certificata agli algoritmi utilizzati. L'idea è quella di utilizzare nuove fonti che soddisfino i seguenti requisiti:

- Abbiano linea editoriale pressochè uguale;
- Siano relative allo stesso Paese;

- Siano in lingua italiana.

I primi due punti sono necessari al fine di massimizzare la sovrapposizione, l'ultimo punto è invece fondamentale per la verifica della pipeline in maniera semplice: si può verificare che anche la traduzione sia effettuata correttamente e che gli accoppiamenti siano consistenti anche rispetto alle notizie visualizzate in lingua originale.

Sulla base di quanto detto in precedenza, sono state aggiunte due fonti a flusso da canali RSS, **ANSA** e **AGI**, per test più accurati, i cui risultati verranno discussi nello specifico nel *Capitolo 5 - Esperimento (AGI vs. ANSA)*.

Questi ulteriori test, che hanno effettivamente confermato la validità della pipeline, hanno quindi portato a due ulteriori quesiti:

- I risultati cambiano disallineando i frame temporali?
- I risultati cambiano modificando le fonti RSS?

Per rispondere a queste ulteriori domande si sono quindi scelte due diverse strade, poi combinate tra loro, ovvero quella del metodo **Pivot - Time-Extended**, o dei **Cluster Temporali**, e quella dei **Feed Cluster**.

I **Cluster** sono definiti come insiemi di notizie raggruppate in base a un determinato criterio, ad esempio le fonti da cui sono reperite (**Feed Cluster**) o la vicinanza temporale (**Cluster Temporale**).

Al concetto di Cluster Temporale si legano i concetti di notiziario **Pivot** e notiziario **Time-Extended**:

- **Pivot**: notiziario che, in un contesto di clustering temporale, è preso in esame solo in un dato frame temporale per tutte i confronti;
- **Time-Extended**: notiziario che, in un contesto di clustering temporale, è soggetto ad un aumento dinamico del suo frame temporale preso in esame, al fine di confrontare notizie da esso pubblicate in un lasso di tempo maggiore rispetto al *Pivot*.

I Cluster Temporali si pongono come obiettivo quello di analizzare le variazioni di esclusività delle notizie del *notiziario A*, modificando dinamicamente il numero di frame temporali del *notiziario B*, come schematizzato in *Figura 3.13*.

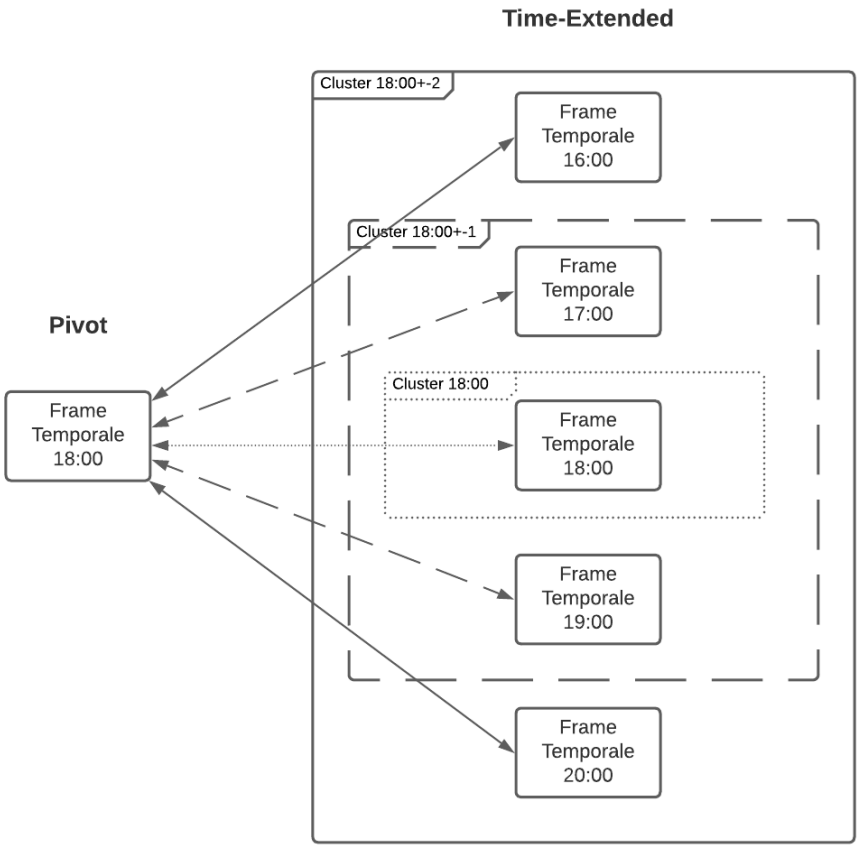


Figura 3.13: Rappresentazione logica del metodo Pivot - Time-Extended: la linea punteggiata indica il frame base, quella tratteggiata indica il Cluster ±1, quella intera indica il Cluster ±2

Questo tipo di analisi ha permesso di valutare le differenze di esclusività nel tempo, ma la sostanziale differenza con le analisi precedenti (vedasi *sezione 3.4*) è che mentre prima si parlava di frame temporali allineati (es.

Notizie delle ore 18 comparate tra loro), con questi confronti, invece, uno dei due notiziari viene analizzato aumentandone la dimensione dei frame ad ogni passo. Il confronto in oggetto fornisce una panoramica sugli argomenti che un notiziario tratta con anticipo o ritardo rispetto all'altro preso in esame, permettendo di valutarne la tempestività di pubblicazione rispetto alle notizie di *Ultim'Ora*, ad esempio.

Per quanto riguarda invece i Feed Cluster, il bisogno di tali raggruppamenti si è manifestato in seguito a controlli manuali dei Feed RSS presi in esame: tali Feed risultavano spesso incongruenti rispetto alla categorizzazione dei Feed stessi, per quanto riguarda le fonti considerate all'inizio dello sviluppo del progetto. Per questo, l'idea è quella di raggruppare canali RSS di categorie differenti per considerare anche notizie che sono state, volontariamente o meno, pubblicate su canali diversi, pur trattando lo stesso argomento, dipendentemente dalla linea editoriale dei notiziari comparati. Uno schema logico riassuntivo dell'idea dietro il Feed Clustering è visibile in *Figura 3.14*.

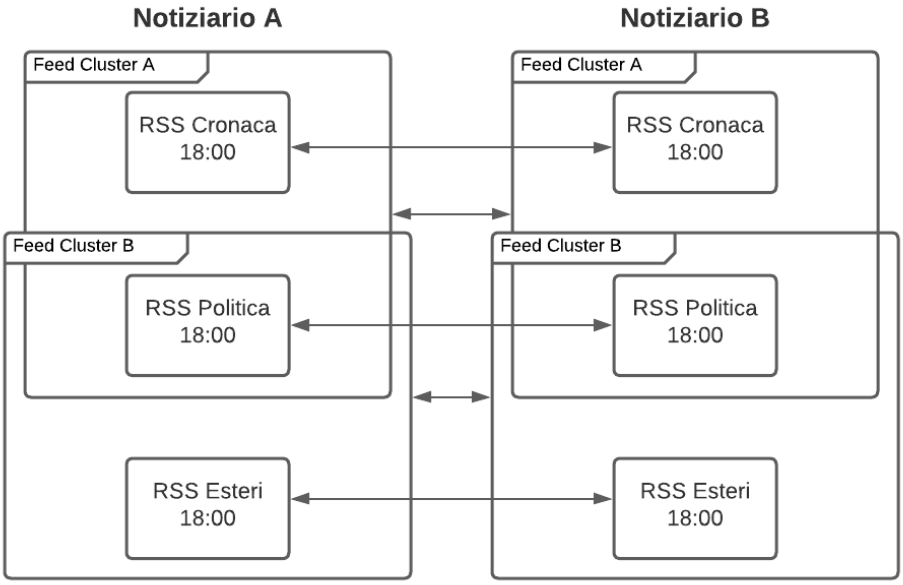


Figura 3.14: Rappresentazione logica del funzionamento dei Feed Cluster

Come rappresentato dallo schema precedente (*Figura 3.14*), nel nostro caso i Feed RSS rilevanti risultano essere tre:

- Cronaca
- Politica
- Esteri

Per poter effettuare un controllo tra diverse categorie, sono stati creati dei cluster tra Cronaca e Politica e tra Politica e Esteri, unendo le coppie di insiemi che intuitivamente sembrano correlati. Inoltre, sono stati confrontati anche cluster di tutti e tre i Feed RSS per confermare il sospetto che una tale unione avrebbe causato bassa percentuale di similarità dovuta all'alta presenza di notizie poco rilevanti per l'uno o per l'altro notiziario.

I risultati di queste ulteriori analisi saranno valutati nel *Capitolo 5 - Esperimento (AGI vs. ANSA).*

3.6 Ulteriori Informazioni Raccolte

Durante la fase di traduzione e analisi delle notizie (vedasi *sezione 3.3.2*) sono state ottenute delle informazioni ulteriori, non necessarie al fine delle valutazioni descritte in precedenza, ma che risultano tuttavia interessanti. Si tratta di entità ottenute tramite NER[8], che rappresentano appunto entità specifiche quali istituzioni governative, Stati o personaggi pubblici, utili per analisi contenutistiche.

Un'altra informazione ottenuta in questa fase è la sentiment analysis[4] di ogni notizia raccolta e tradotta. Il grafico qui riportato (*Figura 3.15*) mostra la variazione della sentiment analysis media per i due notiziari presi in esame in una data giornata.

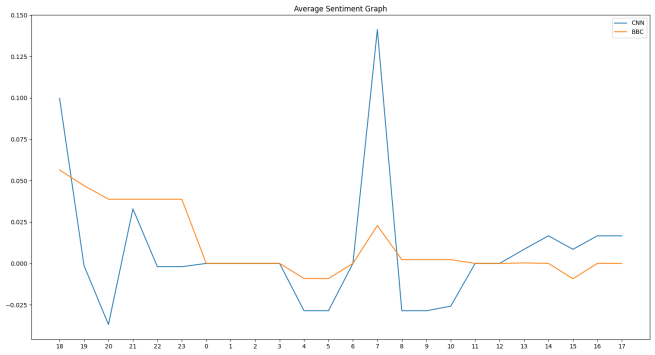


Figura 3.15: Esempio di confronto di sentiment analysis (BBC vs CNN)

È stata ottenuta, con la stessa libreria utilizzata per la sentiment analysis, anche una subjectivity analysis delle notizie, ovvero il calcolo di un grado di oggettività della scrittura di una data notizia. Qui, in *Figura 3.16*, è riportato un grafico dello stesso confronto visualizzata nel grafico precedente, ma comparando la soggettività media.

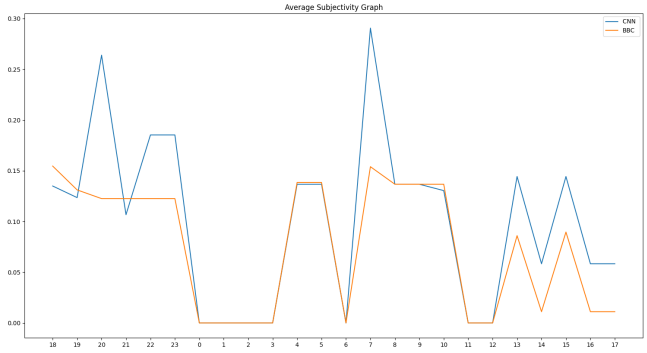


Figura 3.16: Esempio di confronto di subjectivty analysis (BBC vs CNN)

Le idee per l'utilizzo di queste informazioni supplementari verranno presentate nel *Capitolo Conclusioni*.

Capitolo 4

Implementazione

In questo capitolo verranno esposte le scelte implementative che hanno caratterizzato lo sviluppo del progetto. Verranno quindi riesaminati tutti i passi della pipeline descritti nella *Sezione 3.2* (*Figura 3.2*), ma dal punto di vista del codice scritto e delle librerie utilizzate, al fine di fornire una panoramica non solo logica, ma anche tecnica di TARO.

Le varie fasi implementate sono le seguenti:

1. Raccolta delle notizie (newScraping.py);
2. Fetch Continuo delle testate a flusso (attraverso Crontab);
3. Traduzione (newsTranslator.py);
4. Analisi delle notizie (newsAnalyze.py);
5. Confronto e calcolo delle similarità (newsComparer.py);
6. Visualizzazione dei confronti effettusti (newsVisualizer.py).

Il linguaggio di programmazione scelto è Python, questo a causa delle numerose librerie utili sia per la raccolta dei dati che per la loro analisi, come verrà spiegato nelle prossime sezioni.

Per semplicità, verrà dato maggiore spazio all'implementazione della pipeline per le fonti *a flusso*. Ciò è dovuto soprattutto alla maggiore complessità

di analisi e raccolta di notizie, oltre che all'intuizione per cui si possono vedere le fonti *a edizione* come caso particolare di quelle *a flusso*, con snapshot raccolti una sola volta al giorno e confrontati senza utilizzo di *Cluster*.

4.1 Raccolta delle Notizie

La prima fase della pipeline consiste nella raccolta di informazioni. Come spiegato nel *Capitolo 3 - Sviluppo del Progetto*, le notizie, successivamente analizzate, vengono pubblicate online attraverso due canali:

- Feed RSS;
- Pagine HTML.

Verranno di seguito analizzate le metodologie di raccolta per ciascuna delle due fonti.

4.1.1 Feed RSS

I Feed RSS sono dei canali di pubblicazione di file in formato *XML*, contenenti una sezione di tag di intestazione e una parte composta da un certo numero di notizie. Le notizie presenti dipendono dall'ora e dalla data di pubblicazione: diversi giornali hanno politiche diverse per quanto riguarda quali notizie mostrare in un Feed in un dato momento. L'analisi esposta in questa tesi, tuttavia, considera le scelte di pubblicazione come un elemento fondante dei confronti stessi: le testate si differenziano proprio sulla base sulle diverse decisioni rispetto a cosa pubblicare e quando farlo; saranno tali politiche a costituire la base dei confronti. Inoltre, ogni testata distingue differenti Feed per ogni canale tematico: uno per la politica, uno per gli esteri, uno per lo sport ecc. Di seguito è riportato un frammento del Feed RSS del giornale tedesco Spiegel, relativo al canale Esteri.

```
−<channel>
  <title>DER SPIEGEL - Wirtschaft</title>
  <link>https://www.spiegel.de/</link>
 +<description></description>
  <language>de</language>
  <pubDate>Fri, 17 Jun 2022 17:41:32 +0200</pubDate>
  <lastBuildDate>Fri, 17 Jun 2022 17:41:32 +0200</lastBuildDate>
 +<image></image>
 −<item>
  −<title>
     Infrastruktur in Deutschland: Kritik an Chinas Einfluss auf deutsche Windparks
   </title>
  −<link>
     https://www.spiegel.de/wirtschaft/infrastruktur-in-deutschland-kritik-an-chinas-einfluss-auf-deutsche-
     windparks-a-6be6871b-7dc8-4902-aba1-82d474b53f55#ref=rss
   </link>
  −<description>
     Entscheidende Bauteile für die Windpark-Infrastruktur kommen immer häufiger aus China. Aktuell im
     Fokus: ein Projekt vor Borkum, an dessen Errichtung zwei Tochterfirmen eines Pekinger
     Staatskonzerns beteiligt sind.
   </description>
   <enclosure type="image/jpeg" url="https://cdn.prod.www.spiegel.de/images/c43ccecd-5599-49e3-
   a134-f44c17f0ec6f_w520_r2.08_fpx31.32_fpy55.jpg"/>
  +<guid></guid>
   <pubDate>Fri, 17 Jun 2022 16:55:00 +0200</pubDate>
  +<content:encoded></content:encoded>
  </item>
 −<item>
  −<title>
     Milliardengarantien sollen Liquidität von Energieunternehmen sichern
   </title>
  −<link>
     https://www.spiegel.de/wirtschaft/unternehmen/milliardengarantien-sollen-liquiditaet-von-
     energieunternehmen-sichern-a-d6300641-c794-4c81-89cd-f61535df50db#ref=rss
   </link>
```


Figura 4.1: Esempio di Feed RSS

Come in *Figura 4.1*, ogni item costituisce un articolo pubblicato dal giornale online in questione: esso è composto da vari campi, di seguito sono riportati i più rilevanti ai fini di confronti eseguiti.

- Title: il titolo della notizia;
- Link: URL dell'intero articolo sul sito della testata;
- Description: la descrizione, una sorta di sottotitolo;
- pubDate: la data e l'ora di pubblicazione.

Per quanto riguarda *pubDate*, tale campo è stato utilizzato per ottenere l'orario di pubblicazione effettivo della notizia: va però chiarito che quando si parla di orari di *snapshot* e *Frame Temporali* non ci si riferisce a tale orario, bensì all'ora di raccolta di queste informazioni. Questa scelta è in linea con quanto chiarito all'inizio di questa sezione, ovvero con la volontà di confrontare non semplicemente le notizie pubblicate da un *giornale A* con quelle

pubblicate da un *giornale B*, bensì di confrontare anche le relative selezioni nei Feed stessi.

Definita la struttura dei file *XML* e quali siano i campi rilevanti per le analisi in oggetto, va chiarito come tali dati vengano immagazzinati e ottenuti. Per ottenere i dati, sono stati sviluppati degli script che, attraverso richieste *HTTP*, ottenessero i file *XML* a cadenza regolare: la libreria utilizzata per facilitare questa operazione è **Scrapy**[1].

Scrapy è un framework open-source che rende più rapido e semplice il web-scraping, ovvero la raccolta di dati a partire da richieste *HTTP* che forniscono file *HTML* o *XML*. Di seguito viene riportata una parte di codice di uno degli script di scraping utilizzati e ne viene spiegata brevemente la logica.

Listing 4.1: Esempio di funzione di parsing

```
def parse(self, response):
    articles = response.css("item")

    titles= []
    subtitles= []
    dates_raw= []
    urls= []

    for article in articles:
        titles.append(article.css("title::text").get())
        subtitles.append(article.css("description::text").get())
        dates_raw.append(article.css("pubDate::text").get())
        urls.append(article.css("link::text").get())

    dates= self.dateFormatter(dates_raw)
```

Scrapy, utilizzando il comando `scrapy crawl` *`nomescript`*, esegue una richiesta GET all'URL scelto per lo scraping. Una volta ricevuta la risposta, essa viene utilizzata come input per la funzione *parse* (*Listing 4.1*) dello script (parametro *response*), quindi la funzione *css* permette di ottenere elementi

[1] https://scrapy.org/

del *DOM* col tag passato come parametro. L'utilizzo di *::text* permette invece di ottenere il vero e proprio contenuto di quel tag come stringa.

A questo punto, sono stati ottenuti molti dei dati necessari per ogni notizia, ma resta da chiarire come si ottiene il vero e proprio contenuto, grazie all'URL di ogni articolo.

Di seguito è riportato un esempio di pagina web di un articolo, raggiunta grazie all'URL presente nel Feed RSS.

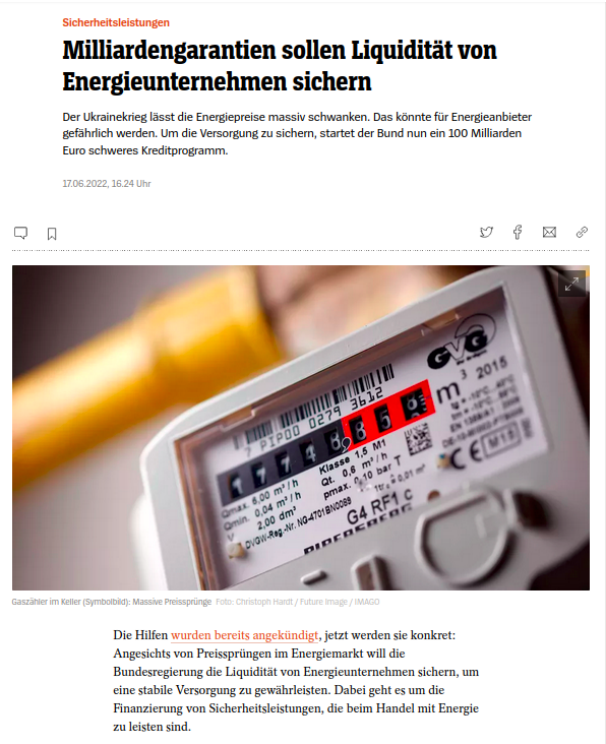


Figura 4.2: Esempio di Articolo di cui fare scraping

Come mostrato in *Figura 4.2*, risulta necessario capire come, a partire dal codice *HTML* della pagina web, si possa ottenere una stringa del contenuto di un articolo. La soluzione è stata analizzare, per ogni fonte, il sorgente delle pagine degli articoli, per capire quali fossero le classi dei tag di contenuto, al fine di poter sfruttare le funzioni descritte precedentemente sul file *HTML*.

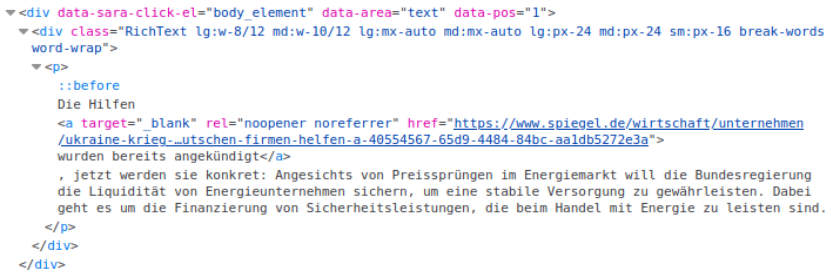


Figura 4.3: Codice sorgente del contenuto di un articolo online

In *Figura 4.3* è possibile vedere come il contenuto degli articoli di Spiegel sia contenuto in un *paragraph* all'interno di un *div* con classe *RichText*: su questo si basa la funzione per il content-scarping, riportata di seguito. Questa funzione prende in input la risposta ad una nuova richiesta *HTTP*, relativa all'URL del singolo articolo, oltre ai dati ottenuti in precedenza e alla lista di notizie già raccolte (*snapshot*).

Listing 4.2: Esempio di funzione di scraping del contenuto

```
def getFullContent(self, rspns, data, snapshot):
        fullcont= rspns.css(".RichText").css("p::text").getall()
        content= ''.join(fullcont)
        item = data
        scraped_info = {
                'title': item[0],
                'date_raw': item[1],
                'date': item[2],
                'url': item['oldurl'],
                'news_url': item[3],
                'subtitle': item[4],
                'content': content,
                'ranked': item['currelem'],
                'placed': 'Abroad',
                'epoch': time.time(),
                'language': 'DE',
                'source': "Spiegel"
        }
        snapshot.append(scraped_info)
```

Nel frammento di codice riportato (*Listing 4.2*), è evidente l'utilizzo della classe *RichText* al fine di ottenere esattamente il contenuto dell'articolo. Successivamente, vengono riempiti i campi relativi ai dati ottenuti in precedenza (come titolo e sottotitolo), aggiungendo anche sia l'URL dell'articolo, che quello del feed. Infine, si aggiunge alla lista di notizie (*snapshot*) l'articolo corrente, fino alla fine dello scraping. La lista di articoli è quindi un file .JSON in output.

Un possibile risultato finale è riportato di seguito.


```
[
    {
        "title": "Impfpflicht: CDU und CSU offen für Kompromiss",
        "date_raw": "Wed, 6 Apr 2022 10:49:09 +0200",
        "date": "2022-04-06",
        "url": "https://www.spiegel.de/politik/index.rss",
        "news_url": "https://www.spiegel.de/politik/deutschland/
        impfpflicht-cdu-und-csu-offen-fuer-kompromiss-a-896afaeb-30a9-41c5-91e0-cfdef996ff6c#ref=rss",
        "subtitle": "Findet die Impfpflicht doch noch eine Mehrheit im Bundestag? Nach SPIEGEL-Informationen hat die Unionsfraktion im
        Gesundheitsausschuss Gesprächsbereitschaft signalisiert – die Ampelparteien wollen das Angebot annehmen.",
        "content": "Nachdem sich Vertreter der Ampelparteien am Dienstag auf ein neues Modell zur Impfpflicht geeinigt haben, signalisiert die
        Unionsfraktion nach SPIEGEL-Informationen nun doch Gesprächsbereitschaft. Damit könnte es am Donnerstag im Bundestag am Ende doch zu einer
        Mehrheit für ein Impfpflicht-Konzept kommen.Der gesundheitspolitische Sprecher der Unionsfraktion, Tino Sorge, hat nach Angaben von Teilnehmern
        im entsprechenden Fachausschuss des Parlaments erklärt, CDU und CSU seien offen für Gespräche über eine gemeinsame Lösung. Mehrere Vertreter der
        Ampelfraktionen, darunter die SPD-Gesundheitspolitikerin Heike Behrens, sollen das Angebot erfreut kommentiert haben. Man werde es annehmen, hieß
        es.",
        "ranked": 12,
        "placed": "Abroad",
        "epoch": 1649275211.6678333,
        "language": "DE",
        "source": "Spiegel"
    },
    {
        "title": "Friedrich Merz: CDU-Chef kritisiert Politik von Karl Lauterbach als »kurzatmig«",
        "date_raw": "Wed, 6 Apr 2022 09:31:20 +0200",
        "date": "2022-04-06",
        "url": "https://www.spiegel.de/politik/index.rss",
        "news_url": "https://www.spiegel.de/politik/deutschland/
        friedrich-merz-kritisiert-karl-lauterbach-kehrtwende-bei-corona-isolationspflicht-a-f0dea486-78b1-4be7-91a0-9b86052a7c2f#ref=rss",
        "subtitle": "Ein zuvor angekündigtes Ende der Isolationspflicht für Coronainfizierte nahm Karl Lauterbach wieder zurück. CDU-Chef Friedrich Merz
        moniert nun den Führungsstil des Gesundheitsministers – und stellt sich gegen eine Impfpflicht ab 60.",
        "content": "Bundesgesundheitsminister  () hat seine vorherige Ankündigung, die Isolation von Coronainfizierten ab dem 1. Mai freiwillig zu
        machen, wieder kassiert. Nach einer entsprechenden -Sendung »Markus Lanz« räumte der Minister auf Twitter einen Fehler ein.Von der Opposition
        gibt es nun deutliche Kritik. Der -Vorsitzende  sagte im , am Kurswechsel Lauterbachs sehe man, wie »kurzatmig« derzeit regiert werde. Beschlüsse
        hätten nicht einmal 48 Stunden Geltung.Lauterbach hatte zuvor auf Twitter mitgeteilt, die Beendigung der verpflichtenden Isolation für
        Coronainfizierte »zugunsten von Freiwilligkeit wäre falsch und wird nicht kommen«. Eine solche Maßnahme entlaste zwar die Gesundheitsämter, sende
        aber ein falsches und schädliches Signal. Corona sei keine Erkältung. »Der Fehler lag bei mir und hat nichts mit der FDP oder Lockerung zu tun«,
        schrieb Lauterbach.",
        "ranked": 14,
        "placed": "Abroad",
        "epoch": 1649275211.5937104,
        "language": "DE",
        "source": "Spiegel"
    }
]
```


Figura 4.4: Esempio di file JSON di output in seguito alla raccolta

Lo script di raccolta dai Feed RSS di ogni testata viene eseguito ogni 15 minuti, al fine di avere snapshot periodici che descrivano accuratamente la scelta delle notizie durante l'intera giornata. Ciò comporta anche una grande quantità di file .JSON da immagazzinare per poter effettuare confronti interessanti. Risulta dunque importante anche la costruzione di un File-

System semplice e intuitivo, ma che non renda difficoltosi i successivi utilizzi dei file nella pipeline.

Il file-system è quindi costituito da una prima distinzione tra testate a flusso e testate a edizione, a loro volta composte da varie directory relative alle diverse lingue dei giornali ispezionati. Un'ultima divisione riguarda le singole testate di ogni lingua: queste ultime directory contengono i singoli file .JSON, con un nome significativo che indichi data, ora ed *epoch* dello snapshot. In *Figura 4.5* un esempio esplicativo della struttura del File-System.

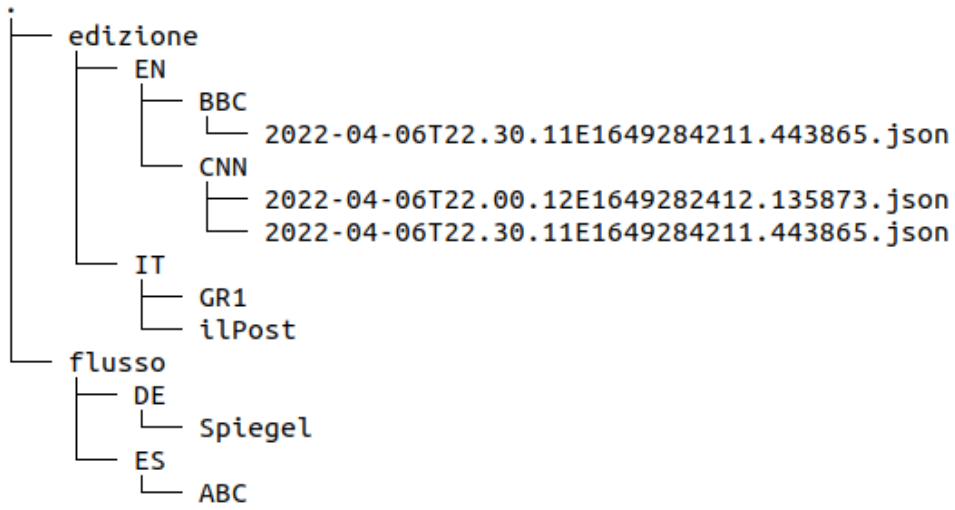


Figura 4.5: Esempio di File-System scelto per la raccolta dei dati

4.1.2 Pagine HTML

Per quanto riguarda le pagine *HTML*, esse seguono un processo di raccolta molto simile rispetto a quello dei Feed RSS, ma risulta necessario ragionare sugli elementi della pagina sin dalla raccolta di informazioni come il titolo o la data di pubblicazione, oltre che per il contenuto. Questa raccolta, infatti, si differenzia dalla precedente poiché risulta necessario fare scraping non solo delle pagine HTML dei singoli articoli, ma anche delle Homepage che ne

contengono un insieme: sono quest'ultime a ricoprire il ruolo di collezioni di notizie precedentemente attribuito ai Feed RSS.

Di seguito viene riportata come esempio la Homepage relativa agli esteri della testata *ilPost*.

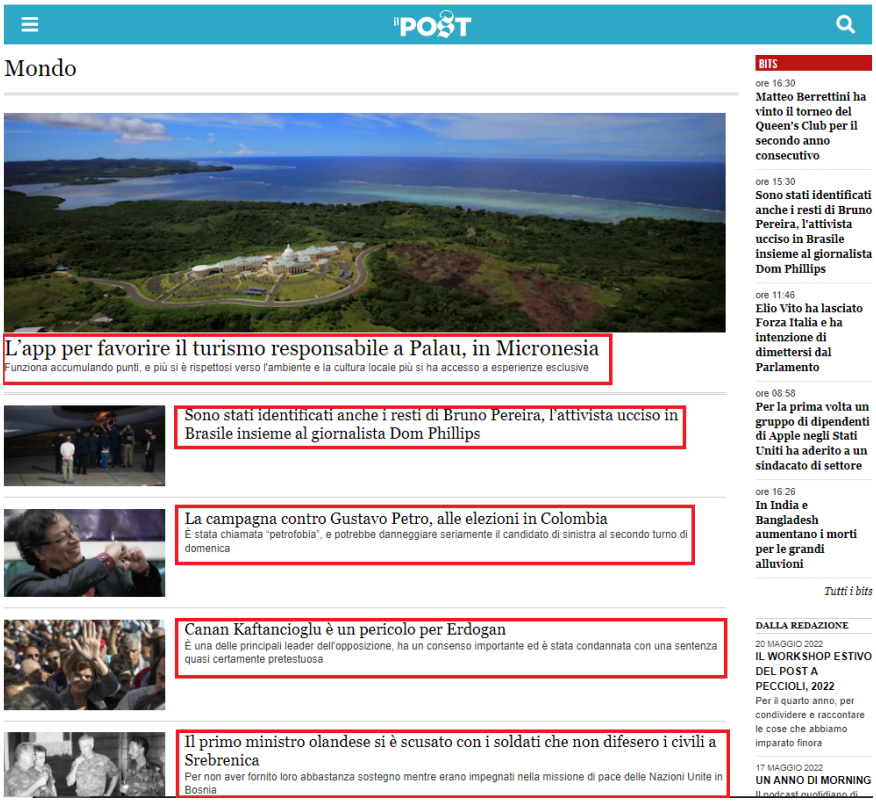


Figura 4.6: Esempio di Homepage di cui effettuare scraping

Come mostrato nella *Figura 4.6*, un primo passo per la raccolta di notizie è quello di capire quali parti della pagina HTML in oggetto identifichino degli articoli di cui fare scraping: nella figura citata, esse sono state evidenziate con un contorno rosso. Da questa Homepage vengono quindi reperite le informazioni di base grazie a Scrapy, seguendo lo stesso principio definito sopra per lo scraping dei contenuti.

A questo punto, il resto del procedimento è del tutto uguale a quello definito per i feed RSS: si effettua un'ulteriore richiesta all'URL di ciascuna

notizia e si ottiene il contenuto come esposto nella *sezione 4.1.1*.

Trattandosi di fonti a flusso, gli snapshot vengono immagazzinati nello stesso File-System definito nella sezione precedente.

4.2 Analisi e Traduzione

Una volta ottenuti gli articoli che verranno in seguito analizzati, è necessario effettuarne la traduzione in lingua inglese, a causa di quanto esposto nella *sezione 3.3.2*.

La libreria scelta per tradurre le notizie è **Argos-Translate**[2], una libreria gratis e open-source che permette traduzioni offline sfruttando il machine learning. Questa libreria permette di installare specifici pacchetti per la traduzione, riducendo anche al minimo l'occupazione di memoria.

Viene di seguito riportato un esempio di file .JSON in versione originale e tradotta.


```
{
    "title": "Draghi: "Usa e Russia siedano al tavolo, Kiev attore principale"",
    "date_raw": "Wed, 11 May 2022 15:24:27 GMT",
    "date": "2022-05-11",
    "url": "https://www.agi.it/estero/rss",
    "news_url": "https://www.agi.it/estero/news/2022-05-11/incontro-draghi-biden-diretta-guerra-ucraina-16688512/",
    "subtitle": "",
    "content": "AGI - È durato un'ora e quaranta minuti l'incontro tra il premier italiano, Mario Draghi, e il presidente degli Stati Uniti, Joe
    Biden;Sul tavolo, le ottime relazioni fra i due Paesi e la conferma della solidità dell'Alleanza atlantica. "La pace sarà quello che vorranno
    gli ucraini, non quello che vorranno altri", ha ricordato il leader italiano durante il meeting trovando la condivsione dell'inquilino della
    Casa Bianca: "Sono d'accordo". Draghi ha proseguito spiegando che "molti in Europa condividono la nostra posizione unita nell'aiutare
    l'Ucraina, e nel sanzionare la Russia. Ma si chiedono anche: come possiamo mettere fine a queste atrocità? Come possiamo arrivare a un cessate
    il fuoco? Come possiamo promuovere dei negoziati credibili per costruire una pace duratura?". E ha concluso: "Al momento è difficile avere
    risposte, ma dobbiamo interrogarci seriamente su queste domande". This afternoon, I met with Prime Minister Mario Draghi of Italy at the White
    House. We reaffirmed the strong and broad partnership between the United States and Italy and underscored our continued commitment to
    supporting Ukraine and imposing costs on Russia. "Abbiamo riaffermato la forte e ampia partnership tra gli Stati Uniti e l'Italia": lo ha
    detto il presidente Usa, Joe Biden, commentando su Twitter l'incontro. "Abbiamo sottolineato - aggiunge - il nostro continuo impegno a
    sostenere l'Ucraina e imporre costi alla Russia"",
    "ranked": 1,
    "placed": "Esteri",
    "epoch": 1652284804.1351998,
    "language": "IT",
    "source": "AGI"
}
```


Figura 4.7: Esempio di articolo in lingua originale

[2]https://github.com/argosopentech/argos-translate

```
{
    "title": "Draghi: \"Usa e Russia siedano al tavolo, Kiev attore principale\"",
    "date_raw": "Wed, 11 May 2022 15:24:27 GMT",
    "date": "2022-05-11",
    "url": "https://www.agi.it/estero/rss",
    "news_url": "https://www.agi.it/estero/news/2022-05-11/incontro-draghi-biden-diretta-guerra-ucraina-16688512/",
    "subtitle": "",
    "content": "AGI - È durato un'ora e quaranta minuti l'incontro tra il premier italiano, Mario Draghi, e il presidente degli Stati Uniti, Joe
    Biden;Sul tavolo, le ottime relazioni fra i due Paesi e la conferma della solidità dell'Alleanza atlantica. "La pace sarà quello che vorranno
    gli ucraini, non quello che vorranno altri", ha ricordato il leader italiano durante il meeting trovando la condivsione dell'inquilino della
    Casa Bianca: "Sono d'accordo". Draghi ha proseguito spiegando che "molti in Europa condividono la nostra posizione unita nell'aiutare
    l'Ucraina, e nel sanzionare la Russia. Ma si chiedono anche: come possiamo mettere fine a queste atrocità? Come possiamo arrivare a un cessate
    il fuoco? Come possiamo promuovere dei negoziati credibili per costruire una pace duratura?". E ha concluso: "Al momento è difficile avere
    risposte, ma dobbiamo interrogarci seriamente su queste domande". "Abbiamo riaffermato la forte e ampia partnership tra gli Stati Uniti e
    l'Italia": lo ha detto il presidente Usa, Joe Biden, commentando su Twitter l'incontro. "Abbiamo sottolineato - aggiunge - il nostro continuo
    impegno a sostenere l'Ucraina e imporre costi alla Russia"",
    "ranked": 1,
    "placed": "Esteri",
    "epoch": 1652284804.1351998,
    "language": "IT",
    "source": "AGI",
    "en_title": "Draghi: "Usa and Russia sit at the table, Kiev main actor"",
    "en_content": "AGI - It lasted an hour and forty minutes the meeting between the Italian premier, Mario Draghi, and the President of the
    United States, Joe Biden. On the table, the excellent relations between the two countries and the confirmation of the solidity of the Atlantic
    Alliance. "Peace will be what the Ukrainians will want, not what others will want", recalled the Italian leader during the meeting finding the
    condivsion of the White House tenant: "I agree."Draghi continued by explaining that "many in Europe share our united position in helping
    Ukraine, and sanctioning Russia. But they also ask: how can we put an end to these atrocities? How can we get to a ceasefire? How can we
    promote credible negotiations to build lasting peace?" And he concluded: At the moment it is difficult to have answers, but we must seriously
    question these questions." "We reaffirmed the strong and broad partnership between the United States and Italy": President Joe Biden said,
    commenting on Twitter on the meeting. "We have stressed - he adds - our continuous commitment to support Ukraine and impose costs on Russia"",
    "en_subtitle": "",
}
```


Figura 4.8: Esempio di articolo tradotto

Come si può notare in *Figura 4.8*, le notizie sono tradotte aggiungendo dei campi `en_NomeCampo`, al fine di non perdere le stringhe in lingua originale.

Dopo la traduzione, vengono effettuate delle analisi, come esposto nella *sezione 3.6*, per ottenere informazioni relative alla Named Entity Recognition e alla Sentiment Analysis. Per effettuare questa analisi è stata usata la libreria di NLP **SpaCy**[3]. Tali analisi sono semplicemente dei passaggi da poter aggiungere alla pipeline della funzione `nlp` della libreria in questione. Vengono di seguito riportati, in *Figura 4.9*, esempi di risultati sia della NER che della Sentiment Analysis, relativi al contenuto della notizia in *Figura 4.8*.

[3]https://spacy.io/

```
"content_polarity": 0.15388888888888888,
"content_subjectivity": 0.41388888888888886,
"content_NER": [
    {
        "word": "an hour and forty minutes",
        "start_char": 16,
        "end_char": 41,
        "label": "TIME",
        "info": "Times smaller than a day"
    },
    {
        "word": "Italian",
        "start_char": 66,
        "end_char": 73,
        "label": "NORP",
        "info": "Nationalities or religious or political groups"
    },
    {
        "word": "Mario Draghi",
        "start_char": 83,
        "end_char": 95,
        "label": "PERSON",
        "info": "People, including fictional"
    },
    {
        "word": "the United States",
        "start_char": 118,
        "end_char": 135,
        "label": "GPE",
        "info": "Countries, cities, states"
    }
]
```


Figura 4.9: Esempio di analisi effettuate sul contenuto di un articolo

Dopo questa fase, le notizie possono essere comparate e ne può essere deciso un grado di similarità utilizzando gli algoritmi scelti per questo progetto.

4.3 Confronto delle notizie

Ottenuti i file .JSON analizzati e tradotti, è possibile utilizzare l'algoritmo **TermMatching** e l'algoritmo del **Coseno** per effettuare i confronti desiderati. I passi per l'implementazione delle comparazioni sono i seguenti:

1. Creazione degli insiemi di cui fare matching;
2. Utilizzo degli algoritmi di confronto;
3. Output dei file di comparazione.

Per quanto riguarda le notizie da fonti *a edizione*, il primo passo risulta banale: basta utilizzare i file .JSON delle due testate prese in esame relative al giorno che si vuole considerare.

Per le notizie *a flusso*, invece, bisogna tenere in considerazione quanto esposto nel *Capitolo 3 - Sviluppo del Progetto* riguardo Frame e Cluster Temporali, oltre che Feed Cluster.

Per quanto riguarda i Frame Temporali, è stato sufficiente utilizzare le informazioni contenute nel nome dei file .JSON per risalire all'orario di scraping: sono state quindi create liste di notizie contenenti gli articoli ottenuti durante il Frame Temporale in esame, una e una sola volta (anche in caso di duplicati tra più snapshot).

Listing 4.3: Funzione di raccolta delle notizie dai file .JSON

```
def multi_news_getter(newsp, start_hour, date):
    got_titles= []
    final_hour = start_hour + HOUR_RANGE

    multi_path = []

    for subdir in os.scandir(f"{BASE_DIR}/{newsp}"):
        filename = subdir.name
        scrape_date = filename.split("T")[0]
        scrape_time = filename.split("T")[1].split("E")[0]
        scrape_hour = int(scrape_time.split(".")[0])

        if scrape_date == date:
            if scrape_hour in range(start_hour, final_hour):
```

```
                multi_path.append(filename)

    to_ret = []
    for single_path in multi_path:
        to_ret = news_getter(f"{newsp}/{single_path}", to_ret,
            got_titles)
    return to_ret
```

La funzione riportata (*Listing 4.3*), chiamata `multi_news_getter` prende come parametri la directory di una testata, l'ora di inizio e la data da considerare per i confronti, restituendo una lista di articoli ottenuta dai .JSON degli snapshot. Per prima cosa, la funzione definisce il Frame Temporale utilizzando la costante `HOUR_RANGE`, successivamente, per ogni file di snapshot della testata in esame, controlla se la data e l'orario di scraping rientrino nei parametri. A questo punto, viene aggiunto il relativo path a una lista di stringhe, che verrà utilizzata per ottenere tutti gli articoli grazie alla funzione `news_getter`, riportata di seguito.

Listing 4.4: Funzione di raccolta delle notizie da un singolo file .JSON

```
def news_getter(subdir, to_append, got_titles):
    to_get_dir = f"{BASE_DIR}/{subdir}"
    to_get = {}
    with open(to_get_dir, "r") as f:
        try:
            to_get = json.load(f)
        except:
            raise Exception("Could␣not␣read␣file␣from␣the␣given␣
                directory:␣" + to_get_dir + ".")
    for single_news in to_get:
        if not single_news['en_title'] in got_titles:
            got_titles.append(single_news['en_title'])
            to_append.append(single_news)
    return to_append
```

La funzione `news_getter`, riportata nel *Listing 4.4*, prende come parametri un path, una lista di articoli (memorizzati come dizionari Python) e una

lista di titoli, utile per assicurare più velocemente l'assenza di duplicati nella lista di notizie. Viene quindi restituita in output la stessa lista di articoli, con l'aggiunta di quelli raccolti dallo snapshot `subdir`, a meno di errori nella lettura del file .JSON.

Vanno inoltre considerate le modifiche effettuate per l'organizzazione in Feed Cluster e Cluster Temporali. Per quanto riguarda i primi, è bastato utilizzare come parametri non il singolo path di una testata, bensì una lista di percorsi, uno per ogni Feed da includere nel Cluster, mantenendo uguale il resto del codice. I Cluster Temporali sono stati invece costruiti aggiungendo una condizione all'if-then della funzione `multi_news_getter`.

Listing 4.5: Condizione per la creazione di Cluster Temporali

```
TIME_EXTENDED in newsp and scrape_hour in range(start_hour -
    CURRENT_SLICE, start_hour + CURRENT_SLICE)
```

La prima condizione visibile nel *Listing 4.5* controlla che la testata di cui si stanno raccogliendo gli articoli sia effettivamente la cosiddetta **Time-Extended** (*sezione 3.5*), mentre la seconda condizione verifica che l'ora dello scraping non sia semplicemente inclusa nel Frame Temporale, ma nell'attuale dimensione del Cluster Temporale, attraverso la costante `CURRENT_SLICE`, che viene aumentata manualmente per effettuare tutti gli esperimenti valutati nel *Capitolo 6 - Esperimento*.

Una volta ottenute le liste di articoli per entrambe le testate in esame, è possibile effettuare i confronti utilizzando l'algoritmo **TermMatching** e l'algoritmo del **Coseno**.

Per quanto riguarda le fonti *a edizione*, le notizie relative vengono comparate utilizzando solo l'algoritmo TermMatching, come esposto nel *Capitolo 3*, comparando solo i titoli delle notizie, data l'assenza di un vero e proprio contenuto.

Di seguito vengono analizzati dettagliatamente entrambi gli algoritmi.

4.3.1 TermMatching

L'algoritmo **TermMatching** si basa sull'estrazione di *concetti* dai testi delle notizie sfruttando la funzione di Pos-Tagging della libreria gratuita e open-source **SpaCy**[4], già citata in precedenza per via del suo utilizzo al fine di effettuare NER e Sentiment-Analysis degli articoli.

Il primo passo da eseguire è, dunque, il Pos-Tagging dell'articolo da comparare e la memorizzazione di quelli che sono stati definiti come *concetti* dello stesso, ovvero i nomi comuni e propri all'interno del testo (*sezione 3.4.1*). Viene di seguito riportata la funzione di *concettualizzazione*.

Listing 4.6: Funzione di estrazione dei concetti di un articolo

```
def get_concepts(news):
    to_ret = []
    for field in FIELDS_TO_NLP:
        if field in news:
            doc_title = NLP(news[field])
            for t_token in doc_title:
                if t_token.pos_ in COMP_POS and t_token.text:
                    to_ret.append(t_token.text)
    return to_ret
```

La funzione riportata nel *Codice 4.6* prende in input un articolo e, grazie alla funzione `NLP` della libreria **SpaCy**, ne effettua il Pos-Tagging. La costante `FIELDS_TO_NLP` è una lista di campi da analizzare per i confronti: sono inclusi il titolo, il sottotitolo e il contenuto (in lingua inglese). La costante `COMP_POS`, invece, è una lista di *Parts of Speech* che vengono considerate *concetti*, in questo caso i nomi propri e i nomi comuni. Viene quindi restituita una lista di stringhe, ovvero le parole che identificano i *concetti* dell'articolo.

Una volta ottenuti i concetti di ogni articolo, è necessario effettuare un *matching* tra i concetti degli articoli da comparare, per verificarne la similarità. La condizione riportata di seguito ha come parametri il numero di *concetti*

[4]https://spacy.io/api/tagger

uguali tra le due notizie e la media delle lunghezze delle stesse, calcolata in base al numero di concetti di ognuna.

$$\frac{(news\_length \times SIMIL\_LOWER\_BOUND)}{100} \leqslant simil\_length$$

Figura 4.10: Formula Similarità TermMatching

La formula in *Figura 4.10* mostra la condizione di similarità tra le due notizie: il numero di *concetti* uguali tra i due articoli deve essere maggiore di una data percentuale di concetti, calcolati sulla lunghezza media delle due notizie esaminate. La percentuale `SIMIL_LOWER_BOUND` è del 10%, secondo quanto esposto nella *sezione 3.4.2*.

Il risultato del confronto attraverso l'algoritmo **TermMatching** è un file .JSON formato da informazioni generali sulle comparazioni e da tutte le coppie di notizie esaminate, con i dati relativi alle singole similarità (vedasi *Figura 4.11*).

```
{
    "len_A": 1,
    "len_B": 2,
    "coupled_A": 1,
    "coupled_B": 1,
    "exclusivity_A_percent": 0,
    "covered_A_percent": 50,
    "exclusivity_B_percent": 50,
    "covered_B_percent": 50,
    "covered_both_percent": 66.6,
    "Court EU condemns Italy, violated air quality limits": {
        "The Kremlin ready to return if Finland enters NATO: \"For us it is a threat\"": {
            "are_similar": false,
            "similar_concepts_number": 1,
            "polarity_X": -0.04682539682539683,
            "subjectivity_X": 0.14206349206349206,
            "polarity_Y": 0.051388888888888894,
            "subjectivity_Y": 0.3061868686868687,
            "simil_concepts": [
                "years"
            ]
        },
        "European Court of Justice in Italy for air quality": {
            "are_similar": true,
            "similar_concepts_number": 18,
            "polarity_X": -0.052777777777777785,
            "subjectivity_X": 0.10208333333333335,
            "polarity_Y": -0.0006995884773662559,
            "subjectivity_Y": 0.10959141681363903,
            "simil_concepts": ["Court", "Italy", "air", "quality", "nitrogen", "dioxide",
            "obligations", "respect", "limit", "value", "areas", "June", "measures", "NO2"
            ]
        }
    }
}
```


Figura 4.11: Esempio di output dell'algoritmo TermMatching: i simil_concepts sono stati tagliati per motivi di visualizzazione

Come visibile nella figura precedente, i primi campi del file .JSON di comparazione sono relativi a valori come il numero di notizie di entrambe le testate, il numero di articoli accoppiati e le percentuali di esclusività e copertura. Sono poi presenti dei campi, indicizzati con i titoli degli articoli, relativi a tutte le coppie di notizie, con un campo `are_similar` booleano, che definisce la similarità delle due notizie, e un campo relativo alla lista di *concetti* raccolti da entrambe, oltre che i valori di polarità e soggettività ottenuti grazie alle funzioni di Sentiment-Analysis.

4.3.2 Similarità Coseno

L'algoritmo del **Coseno** si basa su vettori di occorrenze, ovvero vettori che indicizzano ogni parola del testo e la relativa ricorrenza: tale algoritmo è stato utilizzato anche nello studio di R.Berlanga-Llavori[10] per calcolare la similarità semantica tra documenti con risultati soddisfacenti. La formula in sé, descritta nella *sezione 3.4.3*, ha come obiettivo quello di dare meno peso alle parole particolarmente ricorrenti (come le *stop-words*), al fine di stabilire la similarità tra due testi comparando solo le parole davvero rilevanti: per questo motivo, identificare determinate parole come degli equivalenti dei `simil_concepts`, utilizzati nell'algoritmo **TermMatching**, risulta poco interessante.

L'algoritmo del **Coseno** è stato implementato in maniera nettamente meno verbosa rispetto al **TermMatching**, sfruttando la libreria **sklearn**[5].

Listing 4.7: Calcolo della similarità con Coseno

```
def cosine_comparer(news_A, news_B):
    total_similarity = []
    for field in COSINE_FIELDS_TO_NLP:
        if news_A[field] != "" and news_B[field] != "":
            field_A = news_A[field]
            field_B = news_B[field]
            corpus = [field_A, field_B]
            X_train_counts = count_vect.fit_transform(corpus)
            pd.DataFrame(X_train_counts.toarray(), columns =
                count_vect.get_feature_names_out(), index=['
                field_A', 'field_B'])

            trsfm = vectorizer.fit_transform(corpus)
            pd.DataFrame(trsfm.toarray(), columns = vectorizer.
                get_feature_names_out(), index=['field_A', '
                field_B'])

            similarity = cosine_similarity(trsfm[0:1], trsfm)
            total_similarity.append(similarity[0][1] * 100)
```

[5]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

```
medium_similarity = 0
if len(total_similarity) > 0:
    for single_simil in total_similarity:
        medium_similarity += single_simil
    medium_similarity = medium_similarity / len(
        total_similarity)
return medium_similarity
```

La funzione, mostrata nel *Listing 4.7*, infatti, mostra come le stringhe dei vari campi degli articoli da analizzare vengono trasformate in vettori di occorrenze, per poi calcolarne la similarità attraverso la formula del coseno. Successivamente, viene calcolata la similarità media tra due notizie, rispetto alle similarità dei singoli campi (titolo, sottotitolo e contenuto, in questo caso).

Il valore di similarità viene quindi confrontato con una soglia minima, fissata all'80% per questo progetto, al fine di stabilire se le due notizie comparate trattano lo stesso argomento.

I file di confronto in output di questo algoritmo sono molto simili a quello mostrato in *Figura 4.11* relativo all'algoritmo **TermMatching**, ad esclusione del campo `simil_concepts` lasciato vuoto, poiché, come spiegato precedentemente, potenzialmente fuorviante.

I file .JSON di confronto vengono quindi memorizzati in un File-System del tipo:

`{NotiziarioA}/{NotiziarioB}/{StartHour}-{EndHour}`.json

4.4 Visualizzazione dei Risultati

Ottenuti i file dei confronti voluti, è stato necessario visualizzare i risultati non solo come oggetti JSON, ma anche come dati aggregati in forma grafica: per tale scopo sono stata utilizzate le librerie **Pandas**[6] e **Matplotlib**[7].

[6]https://pandas.pydata.org/docs/index.html
[7]https://matplotlib.org/stable/users/index

Un esempio di utilizzo di Pandas per la visualizzazione è riportato di seguito.

Listing 4.8: Esempio di creazione di un DataFrame per la visualizzazione

```
df = pd.DataFrame(np.array([excl_st, excl_cos]).transpose(),
    index= range(0, 7), columns= [f"Standard Algorithm", f"Cosine
    Algorithm"])
```

Nel frammento di *Listing 4.8* è stata utilizzata la funzione `DataFrame` di Pandas per poi poter visualizzare graficamente i grafici: sono state ottenute due liste di *percentuali di esclusività* da confronti di tipo *Pivot-TimeExtended* effettuate con i due algoritmi di similarità in uso. Gli indici sono quindi di numero pari alla massima estensione della testata *Time-Extended*, in questo caso, massimo +-6 ore.

Listing 4.9: Esempio di plotting di un DataFrame

```
df.plot(title=f"{analyzed} exclusivity percentage", style='o-')
```

Come visibile nel *Listing 4.9*, viene quindi eseguita la funzione `plot`, passando come parametri il titolo del grafico e il tipo di visualizzazione (in questo caso, lineare con cerchi sui valori del DataFrame). Il risultato ottenuto è di seguito riportato.

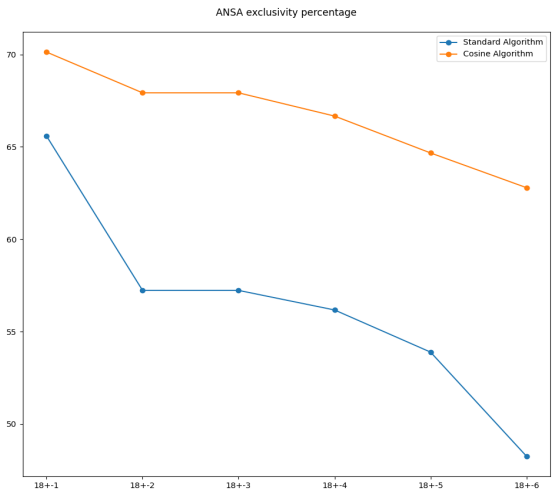


Figura 4.12: Esempio di grafico ottenuto dai file di confronto

Lo stesso procedimento è stato attuato per la visualizzazione dei grafici a barre visibili nel prossimo capitolo, utilizzando, tuttavia, i dati relativi alle cardinalità delle notizie e non le percentuali di esclusività.

Capitolo 5

Esperimento - AGI vs. ANSA

Lo scopo di questo capitolo è mostrare i risultati più rilevanti del progetto e valutarli, per poi poter trarre delle conclusioni sui gradi di esclusività dei notiziari in oggetto, dipendentemente dalle variabili considerate.

5.1 Introduzione

Come esposto nel *Capitolo 3 - Sviluppo del Progetto*, i risultati maggiormente diversificati sono stati ottenuti dal confronto tra le testate *ANSA* e *AGI*, le due maggiori agenzie giornalistiche italiane. La scelta delle fonti è ricaduta su queste due agenzie data la necessità di confrontare due fonti che fossero, intuitivamente, molto vicine in merito agli argomenti trattati: utilizzare, quindi, due agenzie dello stesso stato è risultata una scelta opportuna.

I confronti effettuati hanno come oggetto sia notizie di singoli Feed RSS (Feed Esteri confrontato con Feed Esteri, ad esempio), sia, sfruttando l'idea di clustering, tra insiemi di Feed RSS. Inoltre, sono stati confrontati anche insiemi di notizie non allineati temporalmente, attraverso la tecnica **Pivot - Time-Extended**, che consiste nell'aumentare dinamicamente il numero di ore di pubblicazione preso in esame per una sola delle testate confrontate.

Sono stati fatti confronti, oltre che tra singoli Feed, anche utilizzando i seguenti Feed Cluster:

- Feed Cronaca, Feed Esteri, Feed Politica (*CrEsPo*);
- Feed Cronaca, Feed Politica (*CrPo*);
- Feed Esteri, Feed Politica (*EsPo*).

Non è stato utilizzato un cluster che comprendesse Feed Cronaca e Feed Esteri per la loro poca similarità di argomenti.

Per quanto riguarda i cluster temporali, sono stati effettuati confronti usando vicendevolmente ANSA e AGI come *Pivot* e *Time-Extended*: il *Pivot* è stato fissato alle ore 6, 12 e 18, mentre il *Time-Extended* ha subito un aumento costante di più e meno un'ora a partire dall'ora relativa al *Pivot*, fino a raggiungere un aumento di più e meno sei ore.

Nei grafici mostrati in seguito, la situazione *Pivot-Time-Extended* verrà formalizzata con la forma $\{Ora\_Pivot\}$+-$\{Aumento\_Time - Extended\}$, ad esempio *12+-3*, con ANSA usato come *Pivot* e AGI usato come *Time-Extended*, indica un confronto effettuato tra le notizie pubblicate da ANSA alle 12 e quelle pubblicate da AGI tra le 9 e le 15 il 12/05/2022. Nella prossima sezione saranno prima effettuati confronti tra diversi grafici di similarità, valutandoli in relazione ai canali comparati e alle finestre temporali. Successivamente, saranno analizzati i singoli grafici per ottenere informazioni qualitative sui confronti stessi.

Di seguito verranno riportati principalmente grafici sulla variazione di esclusività della testata *Pivot* rispetto a quella *Time-Extendend*: tale percentuale è calcolata attraverso la formula di seguito riportata.

$$\frac{NotizieA - NotizieAccoppiateA}{NotizieA + NotizieB}$$

Figura 5.1: Formula per il Calcolo dell'Esclusività

5.2 Risultati Ottenuti

Di seguito sono riportati alcuni grafici dai quali è possibile osservare diverse caratteristiche del metodo *Pivot-Time-Extended*: i grafici a barre mostrano la variazione di cardinalità delle notizie, mentre sono successivamente riportati dei grafici che rappresentano la variazione di esclusività dei notiziari.

In *Figura 5.2* è riportato un grafico con la variazione delle varie cardinalità degli articoli: sono riportati il numero di notizie di ANSA e di AGI non accoppiate, oltre alla quantità di notizie coperte da entrambi i giornali, andando a formare il totale delle notizie comparate in quel determinato momento. Tale grafico si riferisce ai confronti tra notizie di ANSA (*Pivot*) e notizie riportate da AGI (*Time-Extended*), a partire dalle ore 6 del 12/05/2022, nel solo Feed RSS di Cronaca. Il grafico a barre evidenzia la differenza nel tempo del numero di notizie per ognuno dei sottoinsiemi: è evidente che il numero di notizie totali non dipenda dall'algoritmo usato e tale valore aumenti non strettamente all'aumento della grandezza del Cluster Temporale. Inoltre, è possibile notare come ANSA pubblichi più notizie di AGI nello stesso canale a prescindere dall'uso come Pivot o Time-Extended: ciò rende particolarmente rilevante un'analisi dell'esclusività nel tempo, posta in forma grafica nella *Figura 5.3*.

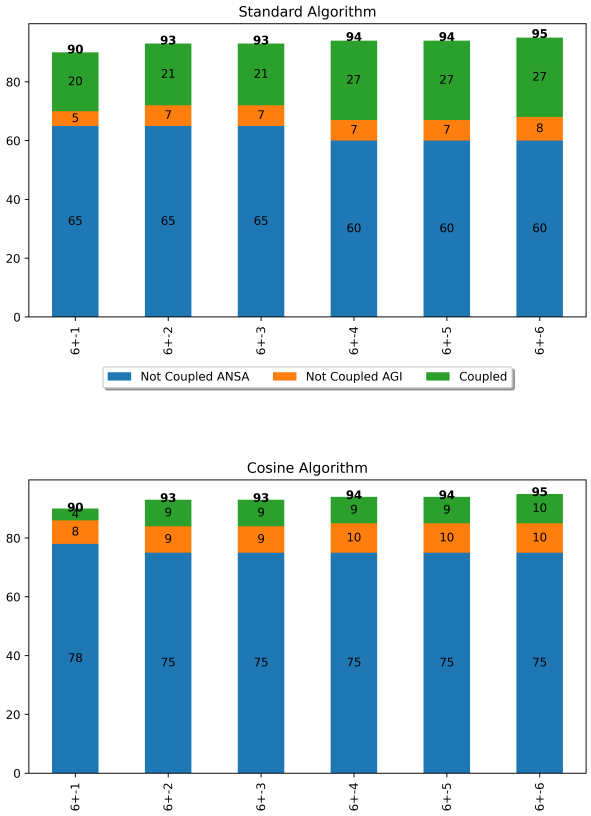


Figura 5.2: Cardinalità notizie di ANSA (Pivot) vs AGI (Time-Extended) dalle 6 del 12/05/2022 (Feed Cronaca). Per cardinalità si intende il numero di notizie di quel tipo raccolte in quel Frame Temporale

Il seguente grafico (*Figura 5.3*) mostra la variazione di esclusività delle notizie di ANSA rispetto alle notizie riportate da AGI, calcolate con la formula in *Figura 5.1*.

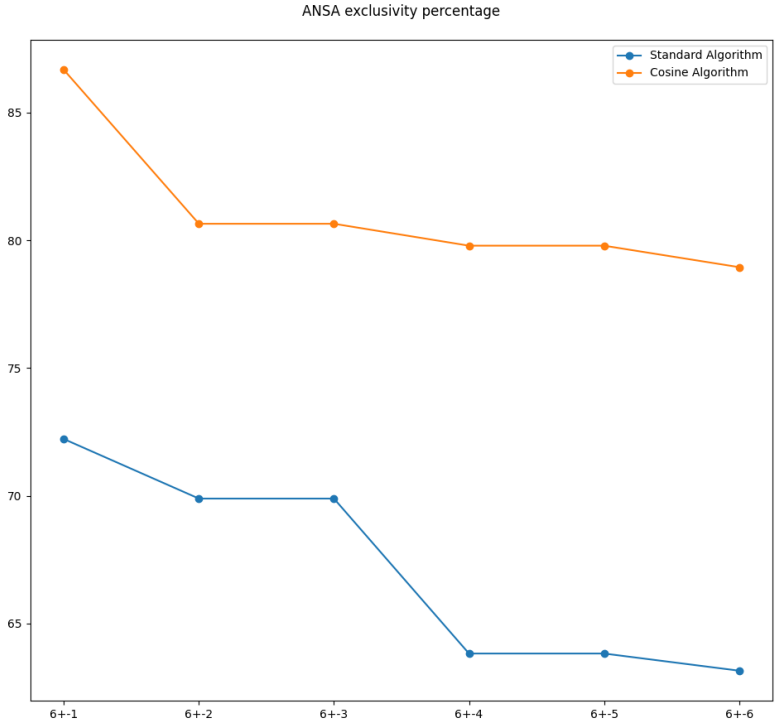


Figura 5.3: Esclusività di ANSA (Pivot) vs AGI (Time-Extended) dalle 6 del 12/05/2022 (Feed Cronaca). Per esclusività si intende la percentuale di notizie trattate solo da una testata, rispetto al totale delle notizie confrontate

Si nota come, all'aumentare dell'intervallo temporale, il numero di notizie coperte esclusivamente da ANSA diminuisce, raggiungendo un plateau a partire da 6±4. Il grafico mostra dei risultati più o meno attesi, in linea con quanto evidenziato nel *Capitolo 3 - Sviluppo del Progetto*: l'*algoritmo del Coseno* accoppia meno notizie, alzando la percentuale di esclusività di ANSA, rispetto a quanto risulta dall'*algoritmo TermMatching*, generalmente meno restrittivo.

Definita quindi la scelta di visualizzazione dei dati, risulta interessante confrontare i confronti stessi.

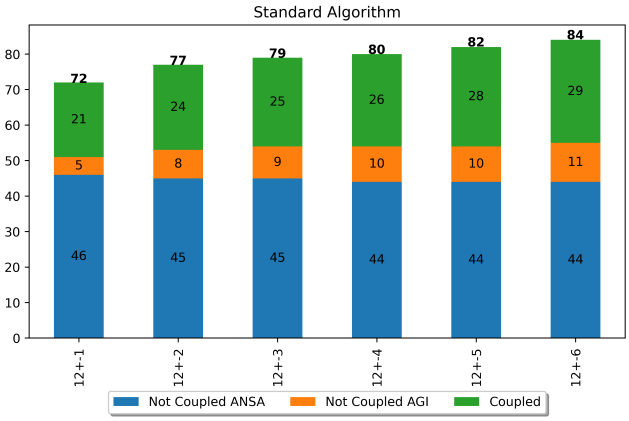
Verranno dunque messe a confronto:

1. Confronti con AGI come *Pivot* e confronti con ANSA come *Pivot*;

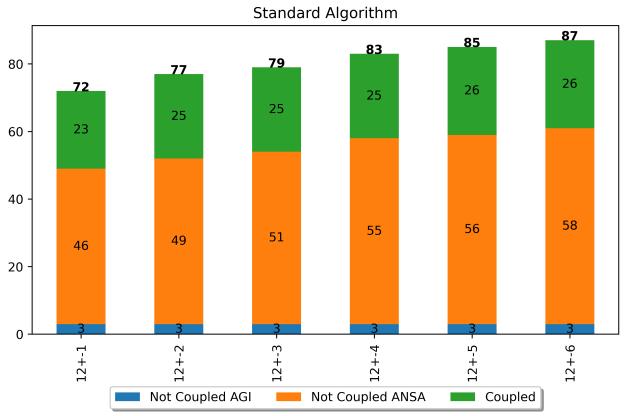
2. Confronti tra differenti Feed Cluster;
3. Confronti tra Cluster e singoli Feed;
4. Confronti relativi a orari differenti.

5.2.1 AGI *Pivot* vs. ANSA *Pivot*

Sono di seguito riportati due grafici a barre relativi ad analisi effettuate a partire dalle 12:00 del 12/05/2022: la fonte Time-Extended copre notizie dalle 6 alle 18, al suo picco. Per semplicità, sono stati considerati solo i risultati dell'algoritmo *TermMatching*.



(a) ANSA Pivot - AGI Time-Extended

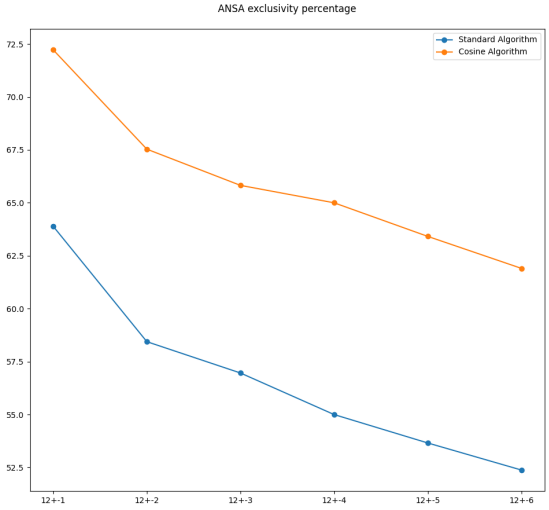


(b) ANSA Time-Extended - AGI Pivot

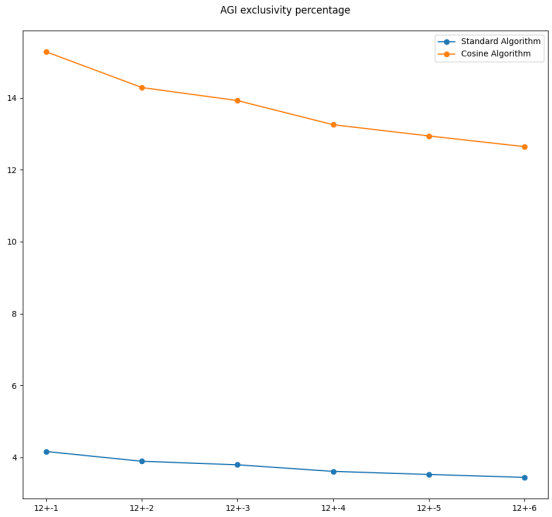
Figura 5.4: Confronto tra cardinalità di AGI Pivot e ANSA Pivot

Dalla *Figura 5.4* è facile notare come usare AGI come Time-Extended permetta di aumentare maggiormente il numero di notizie accoppiate: ciò è dovuto alla maggiore cardinalità di ANSA e la conseguente maggiore ampiezza di tematiche. Infatti, è evidente come ANSA pubblicando più notizie, copra necessariamente più temi rispetto ad AGI e il suo utilizzo come Time-Extended, quindi, non aumenti proporzionalmente il numero di notizie definibili *simili* nel complesso.

Viene quindi riportata di seguito anche la variazione di esclusività relativa agli stessi confronti.



(a) ANSA Pivot - AGI Time-Extended



(b) ANSA Time-Extended - AGI Pivot

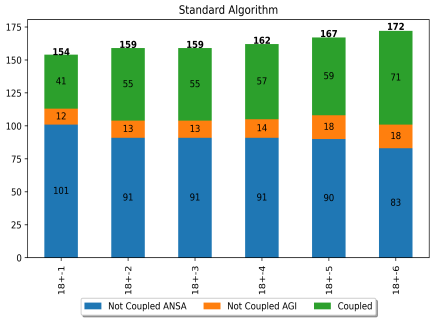
Figura 5.5: Confronto tra esclusività di AGI Pivot e ANSA Pivot

Quanto visibile in *Figura 5.5* è perfettamente in linea con le valutazioni effettuate in precedenza: l'esclusività di ANSA (circa 55%) è percentualmente molto maggiore rispetto a quella di AGI (circa 4%), ma l'utilizzo di AGI come *Time-Extended* (Figura A) permette una decrescita visibilmente più impattante rispetto all'utilizzo di ANSA come tale (Figura B).

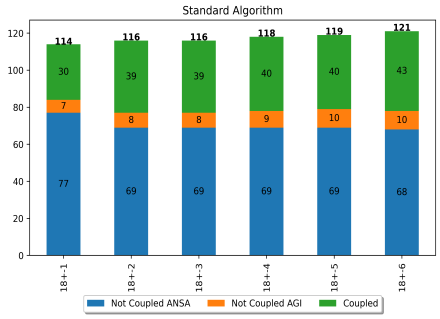
5.2.2 Confronti tra Feed Cluster

In questa sottosezione verranno messi a confronto i grafici di esclusività e le relative cardinalità di diversi Feed Cluster, come definiti nella *Sezione 3.5*.

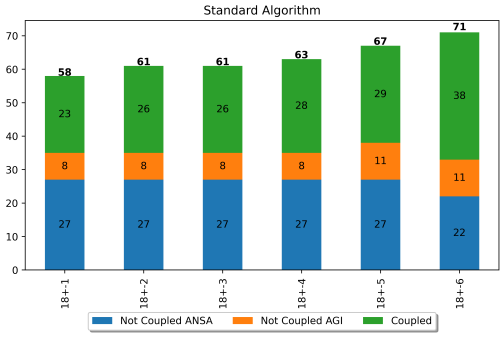
I seguenti grafici mostrano la variazione di cardinalità delle notizie (Accoppiate, esclusive di AGI, esclusive di ANSA) aumentando i Cluster Temporali e utilizzando ANSA come *Pivot* e AGI come *Time-Extended*.



(a) Cronaca-Esteri-Politica



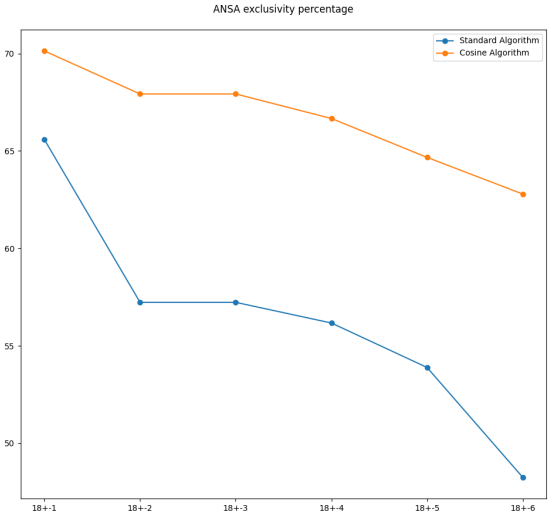
(b) Cronaca-Politica



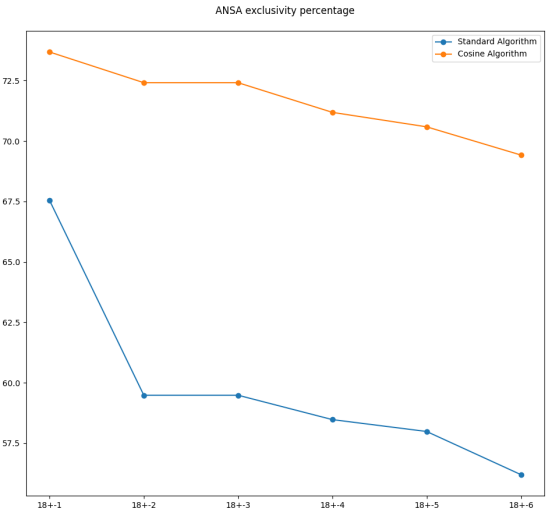
(c) Esteri-Politica

Figura 5.6: Confronto tra cardinalità dei tre Feed Cluster utilizzati

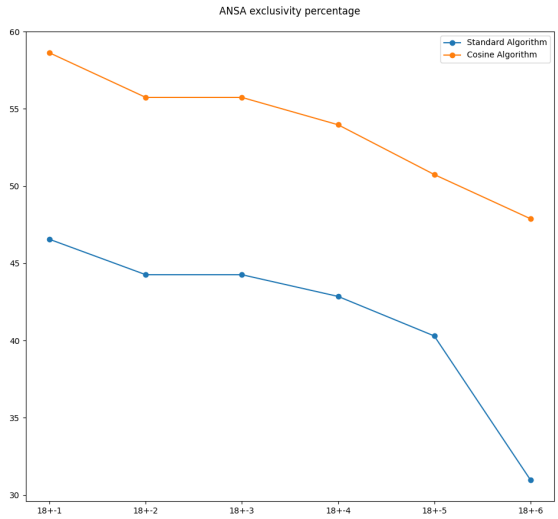
Dai grafici in *Figura 5.6* è evidente come il Feed di Cronaca aumenti considerevolmente la cardinalità totale delle notizie, non portando tuttavia a un proporzionale aumento delle notizie accoppiate: ciò implica che le notizie pubblicate in tale Feed si differenziano maggiormente rispetto alle altre. Questa conclusione è confermata visibilmente dai grafici in *Figura 5.7*.



(a) Cronaca-Esteri-Politica



(b) Cronaca-Politica



(c) Esteri-Politica

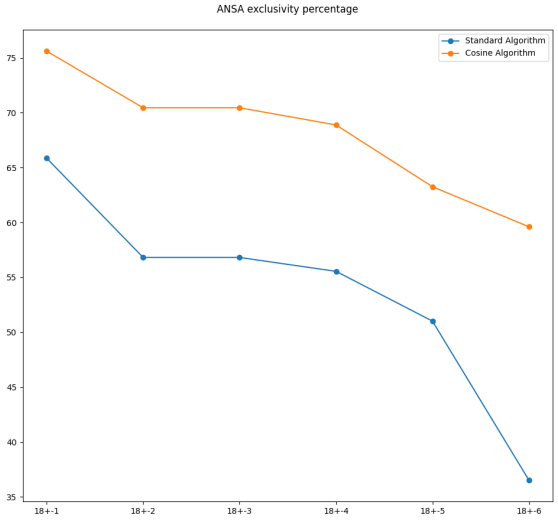
Figura 5.7: Confronto tra variazioni di esclusività dei tre Feed Cluster utilizzati

Dalla *Figura 5.7*, infatti, si nota come l'esclusività delle notizie di ANSA si abbassi da una media del circa 60%, includendo il Feed Cronaca all'interno dei Cluster (Figura A e Figura B), a una del circa 40% per quanto riguarda il Feed Esteri-Politica (Figura C). Questa conclusione attesta delle scelte sostanzialmente differenti per quanto riguarda le pubblicazioni nel Feed Cronaca, comparando le due agenzie, mentre una maggiore somiglianza per quanto riguarda i Feed Esteri e Politica, dimostrando quindi risutati meno interessanti in seguito all'inclusione di tale Feed all'interno dei Cluster.

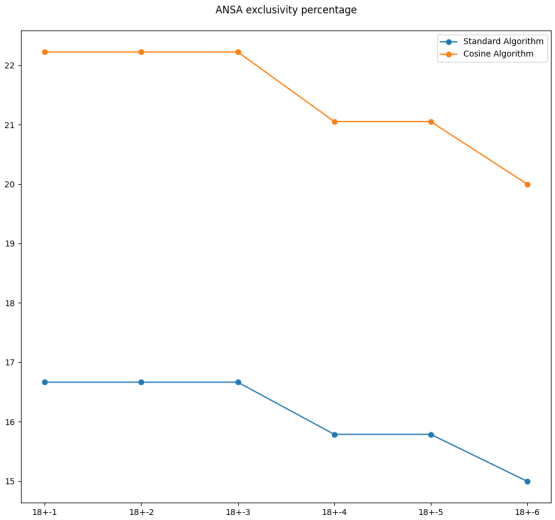
5.2.3 Confronti tra Cluster e singoli Feed

Per quanto rilevato dalla precedente *Sezione 5.2.2*, risulta chiaro come il Cluster Esteri-Politica sia più interessante rispetto ai Cluster Cronaca-Esteri-Politica e Cronaca-Politica, a causa di un disallineamento tra le pubblicazioni nel Feed di Cronaca tra ANSA e AGI.

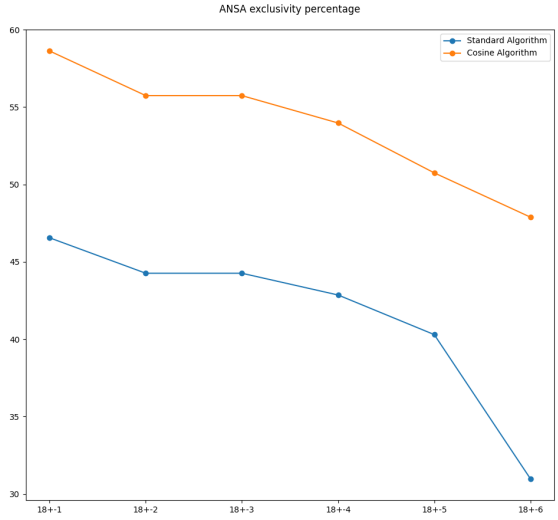
A questo punto, è interessante chiedersi quanto l'utilizzo di un Cluster Esteri-Politica sia interessante rispetto a un confronti tra singoli Feed. Di seguito sono quindi riportati grafici che possano rispondere a tale quesito.



(a) Esteri



(b) Politica



(c) Esteri-Politica

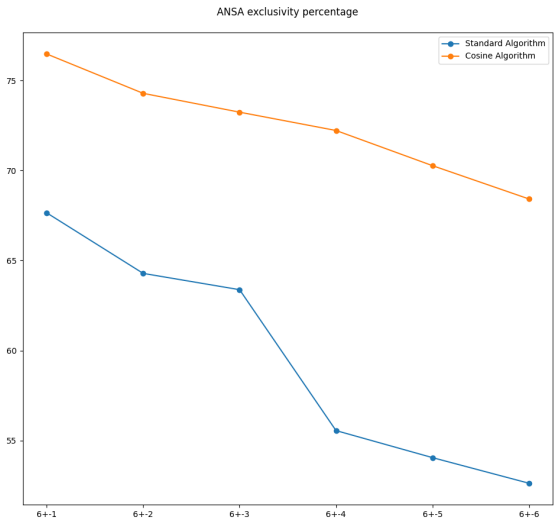
Figura 5.7: Confronto tra variazioni di esclusività dei tre Feed Cluster utilizzati

Dai grafici in *Figura 5.7* risulta chiaro come l'uso del Cluster possa portare a conclusioni poco accurate: la variazione di esclusività risulta molto simile a quella rilevata analizzando il solo Feed Esteri, sia per valori assoluti che per andamento, tuttavia non è lo stesso per quanto riguarda il Feed Politica.

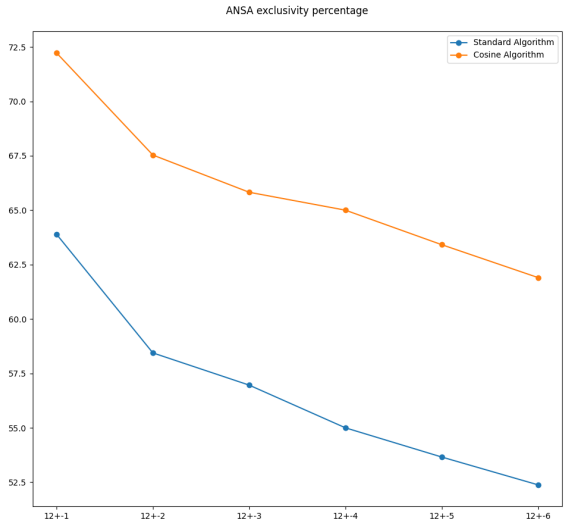
Quest'ultimo, infatti, mostra percentuali di esclusività di ANSA molto più basse e un andamento più costante, risultato "nascosto" nel caso di utilizzo del Cluster. Ulteriori esempi di variazioni di cardinalità ed esclusività di AGI utilizzando Feed Cluster sono riportati nell' *Appendice* (*Figura A.5*, *Figura A.6*).

5.2.4 Confronti tra differenti orari

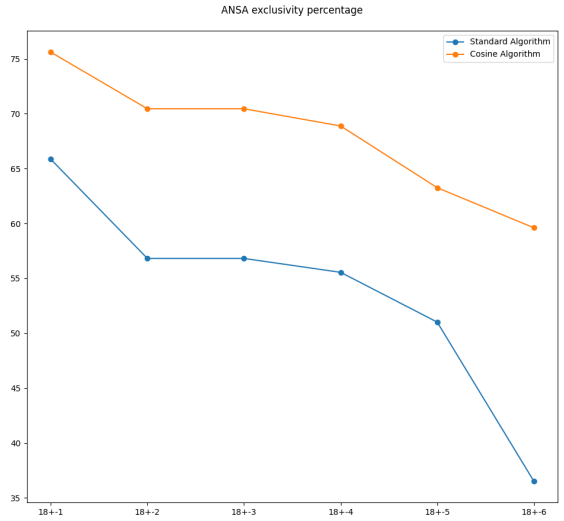
Al fine di confermare una certa consistenza dei confronti, sono stati effettuati i medesimi test con Cluster Temporali (Pivot - Time-Extended) a partire da tre differenti orari: le 6, le 12 e le 18. Di seguito, nella *Figura 5.7* sono riportati i grafici di esclusività di ANSA relativi ai Feed Esteri, per ciascuna delle ore di partenza sopracitate.



(d) Ore 6



(e) Ore 12



(f) Ore 18

Figura 5.7: Confronto tra variazioni di esclusività utilizzando diverse ore di partenza, riguardo il Feed Esteri

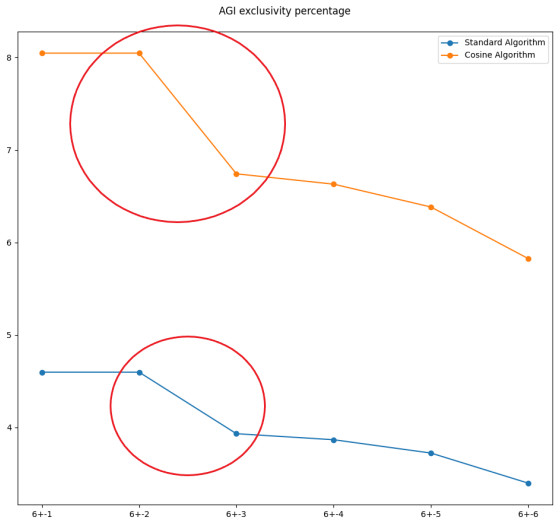
Si può notare un'estrema somiglianza nelle curve del solo Feed Esteri rispetto al Cluster EsteriPolitica, non considerando i valori assoluti I grafici sopra (*Figura 5.7*) mostrano come i valori di esclusività siano simili, ma non uguali: ordini di grandezza comparabili dimostrano una certa consistenza nei

confronti, ma delle variazioni sono comunque interessanti, poiché dimostrano l'importanza di effettuare confronti in diversi momenti della giornata, data la disomogeneità della pubblicazione di notizie. Inoltre, le discese visibili (ad esempio tra il Frame 6+-3 e il Frame 6+-4 in Figura A) implicano probabilmente la pubblicazione di notizie di particolare rilevanza nazionale su entrambi i giornali, ulteriore conferma dell'importanza di valutare le similarità in diversi orari. Altri esempi di grafici di esclusività di AGI per il feed Esteri sono presentati nell'*Appendice* (*Figura A.4*).

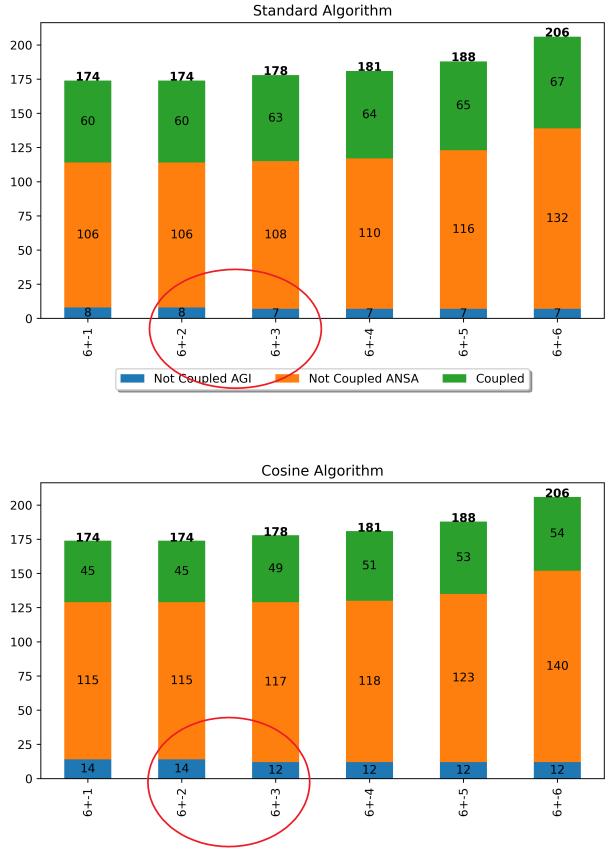
5.3 Valutazioni

I grafici riportati nella precedente sezione rispettano quanto visto anche per le analisi tra fonti disomogenee (*Capitolo 3 - Sviluppo del Progetto*), vanno tuttavia considerate le diverse modalità di calcolo dei risultati: mentre per le fonti analizzate in precedente veniva analizzata la percentuale di notizie coperte da entrambe le testate in esame, per quanto riguarda ANSA e AGI si è deciso di valutare l'esclusività delle testate. Secondo la formula 5.1, il numero di notizie esclusive è normalizzato rispetto al numero di notizie totali analizzate, di entrambe le fonti: tuttavia, tale valore aumenta (non in maniera stretta) necessariamente, secondo la definizione di Cluster Temporale, che implica l'analizzare sempre più notizie riportate dal *Time-Extended*, lasciando invariato il numero di notizie del *Pivot*. Ciò significa che la percentuale di esclusività si abbasserà (o rimarrà invariata) in qualsiasi caso, semplicemente grazie a una crescita del denominatore della formula.

Ciononostante, i grafici risultano ugualmente interessanti identificando non la semplice decrescita dell'esclusività, ma degli specifici decrementi particolarmente rilevanti, come qui di seguito evidenziato, in *Figura 5.8*.



(g) Variazione di Esclusività



(h) Cardinalità delle notizie

Figura 5.8: Evidenziazione decrescita di esclusività

L'aumento del *Time-Extended* da *+-2* a *+-3* comporta un decremento sostanziale della percentuale di esclusività del *Pivot*: l'ipotesi che decrescite particolarmente significative derivino da un effettivo maggior numero di notizie coperte dal *Time-Extended* viene confermata dalla *Figura B*. Nel grafico a barre è infatti visibile come grazie all' espansione del *Time-Extended* sopracitata siano effettivamente aumentate le notizie di AGI coperte anche da ANSA, comparando con entrambi gli algoritmi (vedasi decremento di notizie AGI "non accoppiate").

Quanto esposto sopra, mostra come, attraverso l'utilizzo di Cluster Temporali, aumenti il numero di notizie considerate simili: questo significa che, perlomeno nei casi esaminati, una buona parte di notizie viene coperta da entrambe le fonti, ma non necessariamente nella stessa finestra temporale.

La scelta di ANSA come *Pivot* permette di fare analisi interessanti, tuttavia il suo uso come *Time-Extended* mostra una altissima copertura delle notizie di AGI, inferiori numericamente, ma anche, evidentemente, per quanto riguarda la quantità di tematiche affrontate (come mostrato nella *Sezione 5.2.1*).

Per quanto riguarda i Feed Cluster, come evidenziato in *Sezione 5.2.2* e *Sezione 5.2.3*, è visibile come l'aggregazione di Feed non direttamente allineati porti a una minore accuratezza delle analisi (vedasi il caso del Feed Cronaca). Inoltre, anche l'unione di Feed assimilabili (come Politica ed Esteri) rischia di nascondere informazioni interessanti: questo implica la necessità di un'attenta analisi dei singoli Feed, al fine di decidere la validità di un'eventuale unione degli stessi. Va comunque considerata la natura delle fonti scelte: ANSA e AGI seguono linee editoriali molto simili, compararne quindi i singoli Feed può essere una soluzione ottimale, non è però scontato nel caso in cui si comparino notiziari molto distanti come mostrato nel *Capitolo 3 - Sviluppo del Progetto*.

Le rilevazioni effettuate mostrano quindi come l'intuizione di un disallineamento dei Frame Temporali abbia un riscontro empirico positivo, mentre l'utilizzo dei Feed Cluster non sembra aggiungere informazioni particolarmen-

te interessanti. L'utilizzo delle fonti ANSA e AGI ha reso più semplici le valutazioni sulle tecniche utilizzate e ha permesso di identificare le potenzialità di TARO.

Capitolo 6

Threats to Validity

Quanto esposto nei precedenti paragrafi mostra come attraverso TARO siano stati ottenuti risultati interessanti e soddisfacenti. Inoltre, le potenzialità della pipeline sono estremamente modulabili, come verrà spiegato nel *Capitolo Conclusioni*.

Tuttavia, il progetto non è esente da criticità dovute alle scelte implementative fatte nel processo di sviluppo del software, che saranno analizzate a seguire.

Un limite dell'approccio è dovuto alla soluzione proposta per l'eterogeneità linguistica delle notizie (*sezione 3.3.2*), ovvero la traduzione in lingua inglese: nonostante siano state analizzate in precedenza le differenze semantiche tra testi tradotti in inglese tramite software rispetto alla lingua originale, con risultati soddisfacenti (Pastor et al.[2]), sono state rilevate, durante lo sviluppo di questo progetto, delle differenze nella NER tra testi tradotti e non. La scelta della traduzione in inglese è, infatti, prevalentemente dovuta alla scarsità di librerie e dataset per l'analisi semantica in tutte le lingue prese in esame, senza contare le difficoltà nel rendere modulabile una pipeline eventualmente basata su specifiche librerie per ogni lingua. Ciononostante, sarebbe sicuramente interessante utilizzare strumenti per il calcolo della similarità tra lingue diverse, eventualmente sfruttando anche il machine learning, al fine di limitare le differenze semantiche dovute alla traduzione.

Un'altra potenziale fonte di problemi è costituita dalla scelta dei *concetti* comparati grazie all'algoritmo **TermMatching** (*sezione 4.2.1*): TARO considera come tali i nomi comuni e i nomi propri rilevati nel testo, ma tale scelta potrebbe risultare meno efficace rispetto al confronto, ad esempio, dei verbi presenti nell'articolo.

Un altro aspetto che può drasticamente influenzare il tipo di analisi presentate in questa tesi è la necessità di scegliere delle soglie oltre le quali gli algoritmi utilizzati considerano due notizie come simili: le percentuali minime di concetti uguali o di vicinanza tramite formula del coseno, necessarie per stabilire un valore booleano di similarità tra coppie di articoli, sono state scelte attraverso numerosi test sperimentali e controlli manuali degli accoppiamenti effettuati, ma non si possono escludere errori di valutazione umani durante la verifica.

Va anche evidenziato come la scelta delle fonti (Feed RSS e Homepage) sia stata dettata da esigenze implementative e di tempo: non è accertato che tali canali di pubblicazione siano rappresentativi delle testate prese in esame e che, piuttosto, non ci siano fonti più adeguate.

Conclusioni

Il progetto TARO permette di raccogliere, allineare, analizzare e confrontare insiemi di notizie da più fonti in maniera modulare, sfruttando la similarità semantica per definire percentuali di copertura e di esclusività di testate giornalistiche rispetto ai loro articoli. Sono state scelte ANSA e AGI come fonti per sperimentare le potenzialità e la validità delle analisi di TARO, mostrando come i risultati attesi venissero rispettati: l'algoritmo TermMatching permette di fare confronti meno stringenti e più incentrati sull'argomento dell'articolo piuttosto che sulla specifica notizia, mentre l'algoritmo del Coseno risulta più restrittivo, lasciando ampio margine all'utilizzo di nuovi algoritmi in futuro. Si è mostrato come il numero di notizie di una testata influisca sensibilmente sulla copertura degli argomenti e verificato che l'utilizzo di Cluster Temporali permetta di identificare notizie simili pubblicate ad orari differenti dalle due testate, permettendo di valutare l'utilizzo, in fase di miglioramento della pipeline, di nuovi sistemi di Clustering. Anche la scelta dei canali di pubblicazione ha influito sensibilmente sui risultati, evidenziando come l'estensione di tale insieme possa permettere di effettuare nuove analisi.

Il progetto discusso in questa tesi, oltre ad aver portato a risultati soddisfacenti, lascia anche ampio spazio a miglioramenti e ulteriori applicazioni future: sarebbe possibile aggiungere nuovi passaggi alla pipeline o modificare quelli esistenti, oltre che utilizzare gli output per effettuare particolari analisi. Di seguito sono descritti alcuni dei possibili sviluppi futuri del progetto esposto.

Per questo progetto sono stati utilizzati due algoritmi di similarità: l'al-

goritmo del **Coseno** e l'algoritmo di **TermMatching**, ideato proprio per gli scopi di questa tesi. Tuttavia, sono diversi gli algoritmi possibili già utilizzati per calcolare similarità semantica, ad esempio l'indice di **Jaccard**[5], che serve a calcolare il numero di elementi uguali in due insiemi. Inoltre, non sono stati considerati gli algoritmi di Machine Learning che sfruttano le ontologie per calcolare la vicinanza tra concetti. Sono anche interessanti nuovi approcci al problema, come la soluzione delle *Lexical Chains*, proposta da S.J.Green[3], che sfrutta concetti semantici come le antinomie per definire la vicinanza tra parole.

Un possibile miglioramento per il calcolo della similarità, oltre all'utilizzo di nuovi algoritmi, potrebbe anche essere rappresentato dall'utilizzo del database lessicale **WordNet**[1] per poter normalizzare ogni parola ed evitare casi di sinonimie, non considerate attualmente. WordNet, infatti, funge da dizionario online che identifica parole e ne segnala, ad esempio, iponomi o ipernomi favorendo il calcolo della vicinanza semantica tra parole non strettamente uguali. Va tuttavia considerato lo stato di WordNet stesso, attualmente non completo e spesso poco preciso.

Un modo differente per intendere la pipeline sarebbe possibile attraverso l'utilizzo di **Topic Cluster**: sarebbe possibile utilizzare specifiche tecniche per dividere le notizie per macro-argomenti e quindi effettuare analisi di similarità all'interno di tali sottoinsiemi. Un possibile modello applicabile sarebbe **LDA**[11] per la definizione di Topic tra differenti articoli. Sarebbe anche possibile sfruttare la NER, già effettuata per questo progetto, al fine di definire dei Topic in base alle *entità* citate all'interno degli articoli.

Per quanto riguarda i possibili *utilizzi* del progetto, un esempio lampante è il confronto delle **Sentiment Analysis** degli articoli, valore già calcolato per ogni notizia internamente alla pipeline descritta: sarebbe possibile comparare la polarità media delle notizie tra testate o tra lingue diverse, oltre che le polarità specifiche per determinati argomenti di interesse nazionale o internazionale, al fine di effettuare analisi sociologiche o culturali, ad esempio

[1] https://wordnet.princeton.edu/

seguendo quanto esposto da Agarwal et al.[1].

Una possibile rimodulazione della pipeline potrebbe riguardare la traduzione, evidenziata come criticità nel capitolo precedente *Threats to Validity*. Sarebbe possibile, infatti, scegliere altre librerie che siano costruite nell'ottica di tradurre articoli di giornale, per superare eventuali bias della traduzione, o anche studiare metodi per confronti tra fonti in lingue differenti, come tentato in questo progetto.

Considerato anche quanto esposto nel precedente capitolo, potrebbe risultare interessante la considerazione di fonti di raccolta delle notizie differenti anche internamente ai siti web stessi. Inoltre, potrebbe essere utili unire più canali di pubblicazione, come gli stessi Feed RSS e Homepage.

Il progetto esposto risulta molto promettente e le sue potenzialità sono non solo puramente tecniche, ma anche di studio socio-culturale nell'ambito della ricerca scientifica, con possibili confronti tra testate nazionali e internazionali attraverso l'utilizzo di nuovi algoritmi di analisi semantica.

Appendice A

In questo capitolo di appendice verranno aggiunte diversi grafici utili come integrazione per i capitolo precedenti, relativamente a percentuali di esclusività e copertura, confronti tra sentiment analysis e cardinalità delle notizie confrontate.

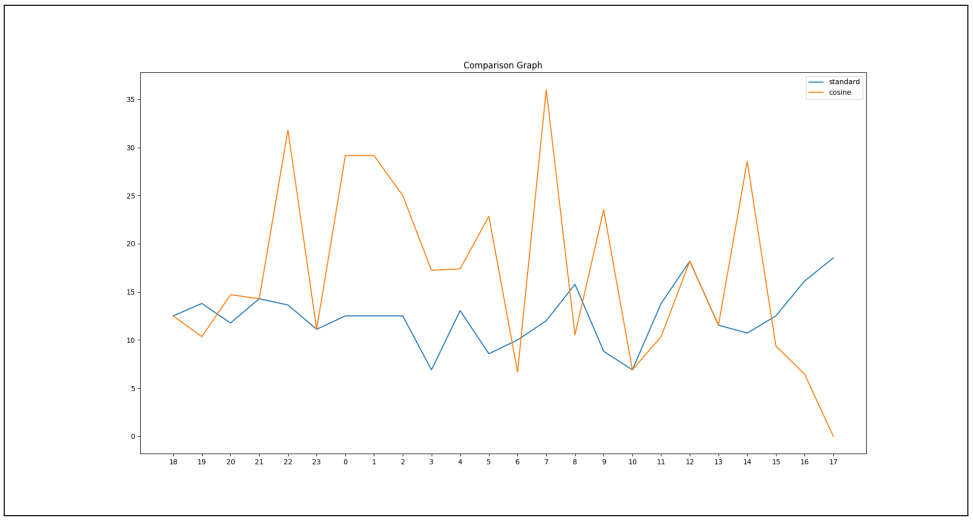


Figura A.1: Grafici di confronto tra Spiegel e France24 il 21/04/2022

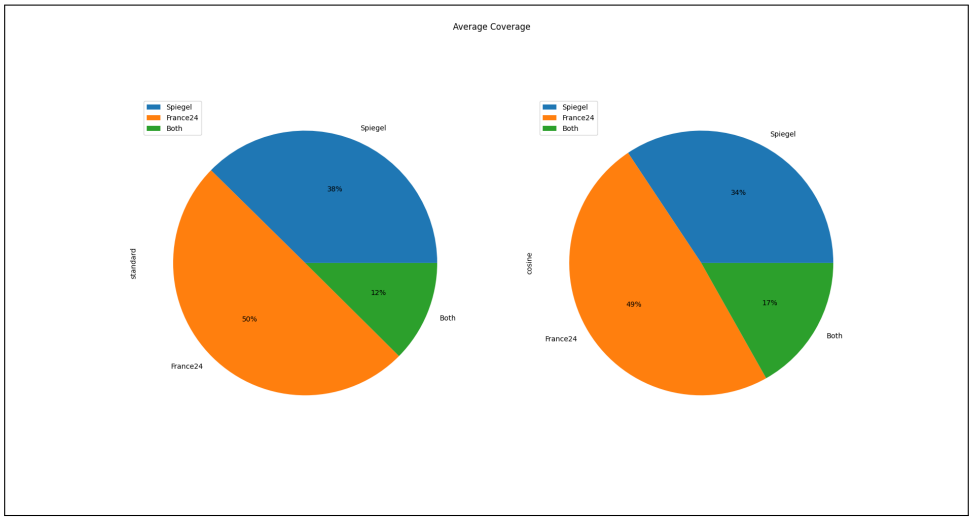
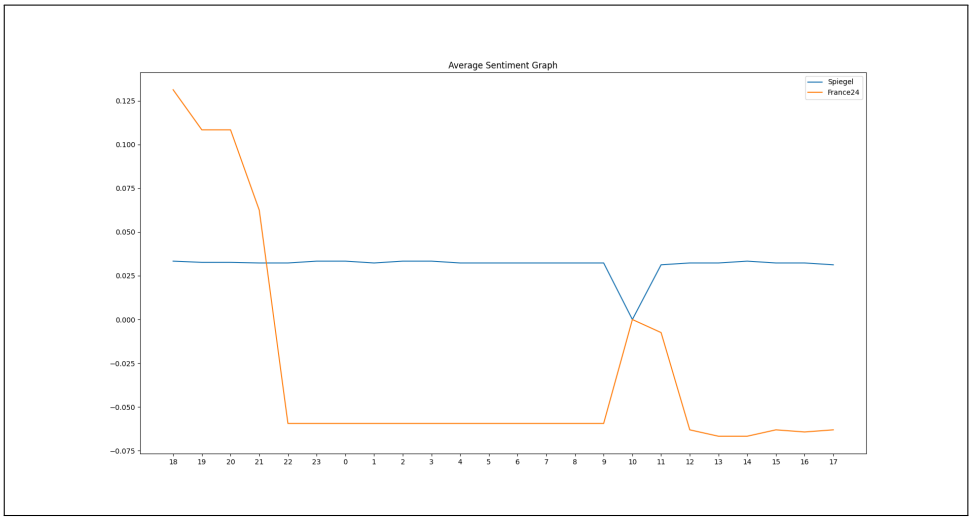
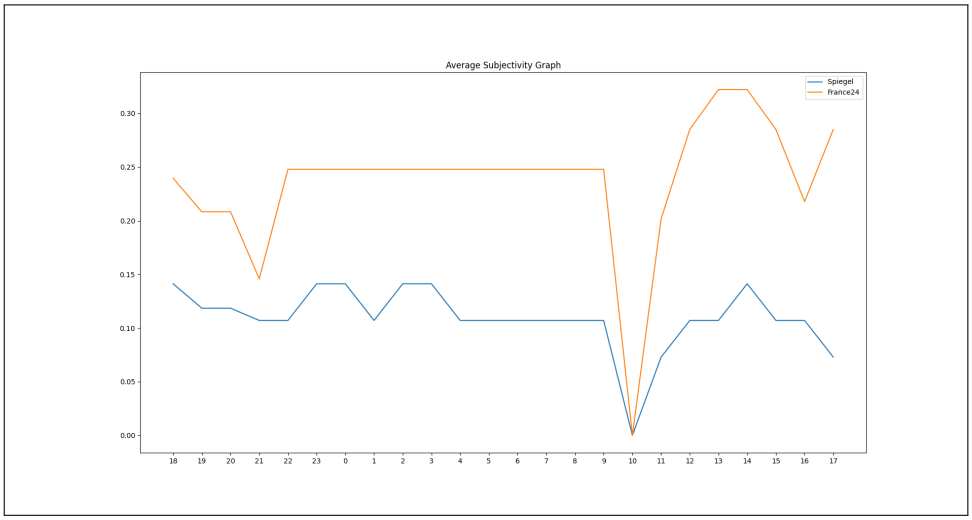


Figura A.2: Grafici a torta di confronto tra Spiegel e France24 il 21/04/2022

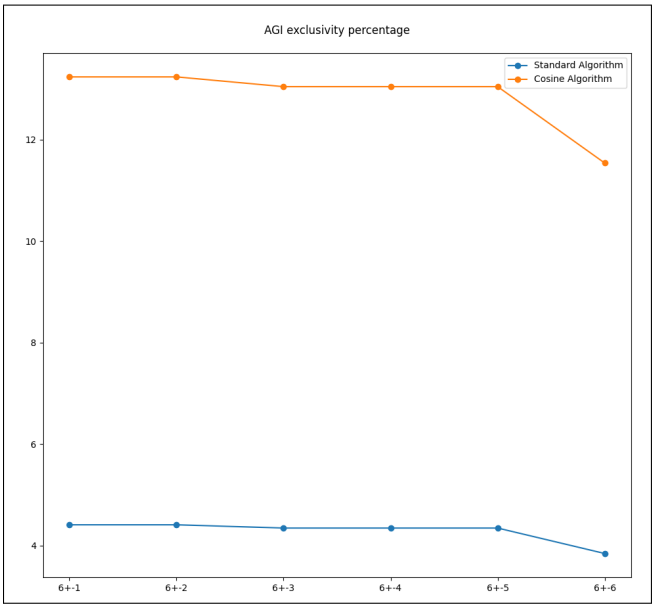


(a) Confronto Sentiment Analysis

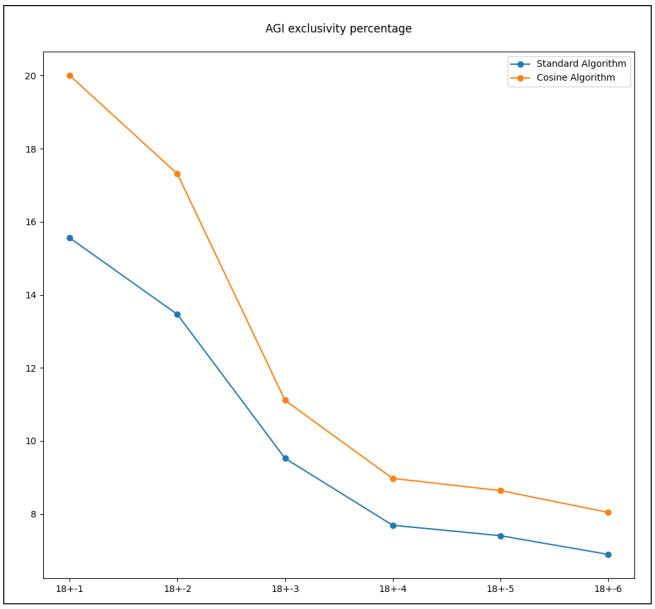


(b) Confronto Subjectivity Analysis

Figura A.3: Grafici di sentimento tra Spiegel e France24 il 21/04/2022

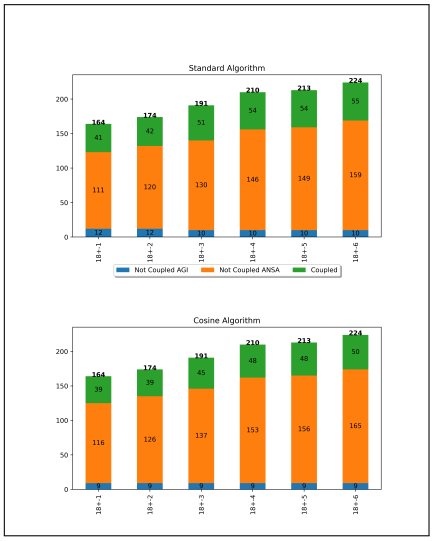


(a) Pivot ore 6

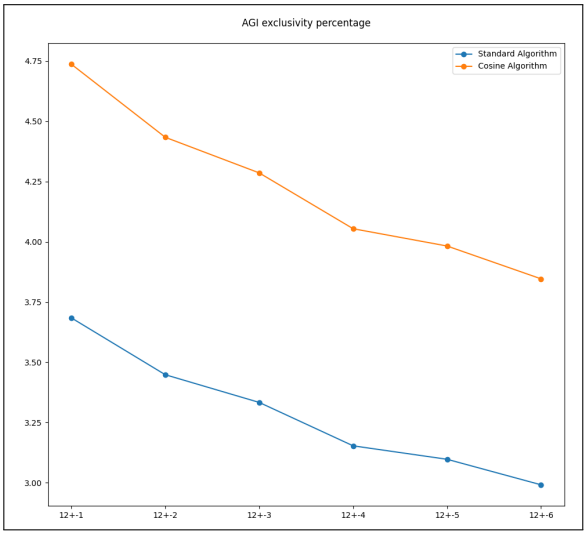


(b) Pivot ore 18

Figura A.4: Grafici di esclusività del solo Feed RSS Esteri per AGI il 12/05/2022

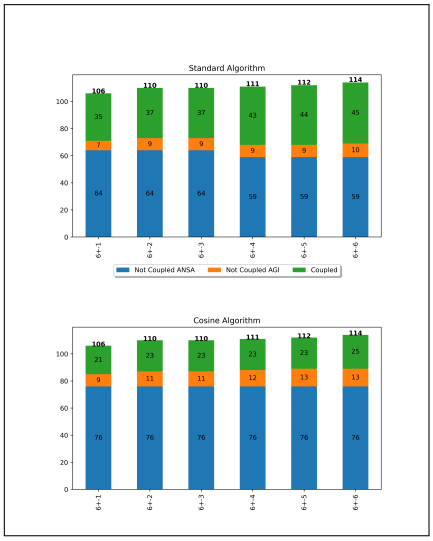


(a) Grafico a barre delle cardinalità

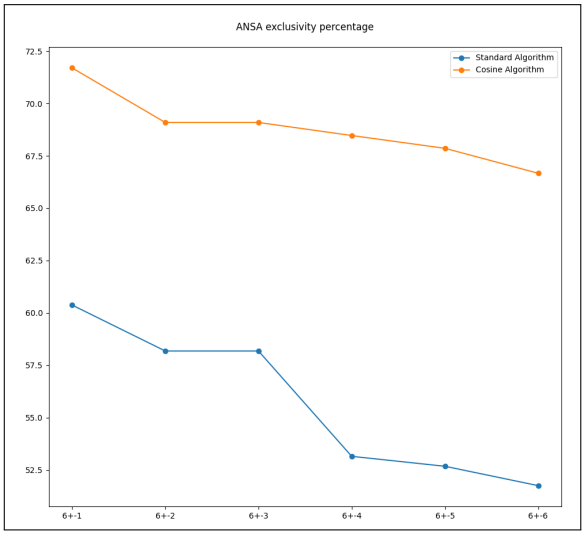


(b) Grafico della variazione di esclusività di AGI

Figura A.5: Esempio di variazione del Cluster CrEsPo usando Cluster Temporale +-6 in data 12/05/2022 tra ANSA e AGI



(a) Grafico a barre delle cardinalità



(b) Grafico della variazione di esclusività di AGI

Figura A.6: Esempio di variazione del Cluster CrPo usando Cluster Temporale +-6 in data 12/05/2022 tra ANSA e AGI

Bibliografia

[1] Apoorv Agarwal et al. “Opinion mining of news headlines using SentiWordNet”. In: (2016), pp. 1–5. DOI: 10.1109/CDAN.2016.7570949.

[2] Gloria Corpas Pastor et al. “Translation universals: do they exist? A corpus-based NLP study of convergence and simplification”. In: (ott. 2008), pp. 75–81. URL: https://aclanthology.org/2008.amta-papers.5.

[3] S.J. Green. “Building hypertext links by computing semantic similarity”. In: *IEEE Transactions on Knowledge and Data Engineering* 11.5 (1999), pp. 713–730. DOI: 10.1109/69.806932.

[4] Felix Hamborg e Karsten Donnay. “NewsMTSC: A Dataset for (Multi-)Target-dependent Sentiment Classification in Political News Articles”. In: (apr. 2021), pp. 1663–1675. DOI: 10.18653/v1/2021.eacl-main.142. URL: https://aclanthology.org/2021.eacl-main.142.

[5] John Hancock. “Jaccard Distance (Jaccard Index, Jaccard Similarity Coefficient)”. In: (ott. 2004). DOI: 10.1002/9780471650126.dob0956.

[6] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari e Noor Akhmad Setiawan. “Cosine similarity to determine similarity measure: Study case in online essay assessment”. In: (2016), pp. 1–6. DOI: 10.1109/CITSM.2016.7577578.

[7] Vinci Liu e James R. Curran. “Words and Word Usage: Newspaper Text versus the Web”. In: *Proceedings of the Australasian Language Technology Workshop 2005* (gen. 2005), pp. 167–175.

[8] David Nadeau e Satoshi Sekine. "A Survey of Named Entity Recognition and Classification". In: *Lingvisticae Investigationes* 30 (ago. 2007). DOI: 10.1075/li.30.1.03nad.

[9] Teresa Paccosi e Alessio Palmero Aprosio. "KIND: an Italian Multi-Domain Dataset for Named Entity Recognition". In: (2021). DOI: 10.48550/ARXIV.2112.15099. URL: https://arxiv.org/abs/2112.15099.

[10] Aurora Pons-Porrata, Rafael Berlanga-Llavori e José Ruiz-Shulcloper. "Temporal-Semantic Clustering of Newspaper Articles for Event Detection". In: (gen. 2002), pp. 104–113.

[11] Jonathan K Pritchard, Matthew Stephens e Peter Donnelly. "Inference of Population Structure Using Multilocus Genotype Data". In: *Genetics* 155.2 (giu. 2000), pp. 945–959. ISSN: 1943-2631. DOI: 10.1093/genetics/155.2.945. eprint: https://academic.oup.com/genetics/article-pdf/155/2/945/42030266/genetics0945.pdf. URL: https://doi.org/10.1093/genetics/155.2.945.

[12] L. Cervi S. Tejedor e F. Tusa. "Information on the COVID-19 Pandemic in Daily Newspapers' Front Pages: Case Study of Spain and Italy". In: *International Journal of Environmental Research and Public Health* 17 (2020). DOI: 10.3390/ijerph17176330.

[13] Nadella Sandhya e A. Govardhan. "Analysis of Similarity Measures with WordNet Based Text Document Clustering". In: (2012). A cura di Suresh Chandra Satapathy, P. S. Avadhani e Ajith Abraham, pp. 703–714.

[14] Hemlata Shelar et al. "Named Entity Recognition Approaches and Their Comparison for Custom NER Model". In: *Science & Technology Libraries* 39.3 (2020), pp. 324–337. DOI: 10.1080/0194262X.2020.1759479.

[15] Vikas Thada e Vivek Jaglan. "Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm". In: *International Journal of Innovations in Engineering and Technology* 2 (ago. 2013), pp. 202–205.

[16] Kadir Yalcin, Ilyas Cicekli e Gonenc Ercan. "An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding". In: *Expert Systems with Applications* 197 (2022), p. 116677. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.116677. URL: https://www.sciencedirect.com/science/article/pii/S0957417422001610.

[17] Niaz Zafri et al. "A Content Analysis of Newspaper Coverage of COVID-19 Pandemic for Developing a Pandemic Management Framework". In: *Heliyon* 7 (mar. 2021), e06544. DOI: 10.1016/j.heliyon.2021.e06544.

Ringraziamenti

La nascita e lo sviluppo di TARO, così come la stesura di questa tesi, non sono stati semplici e non sarebbero stati neppure possibili senza tutte le persone che mi sono state vicine, che sia da qualche mese come da tutta la vita. Ringrazio i prof. Di Iorio e Barabucci, che hanno creduto in quello che ho e abbiamo costruito in questo anno e mi sono stati immensamente d'aiuto con gentilezza e simpatia.

Ringrazio i miei genitori, Romeo e Simona, e mia sorella Virginia, per avermi ascoltato ed avermi rivolto parole di amore anche nei momenti in cui mi sentivo più a terra (sopportandomi da ormai un paio di decenni).

Ringrazio il Dukes, che mi rifiuterò di chiamare Massimo anche in un contesto così ufficiale, per essere stato l'amico più sincero e presente che potessi desiderare.

Ringrazio Elisa per avermi donato affetto e serenità, per avermi capito e supportato in qualsiasi mio percorso sempre con entusiasmo, facendomi sentire abbracciato in tutti i momenti passati insieme. Spero di restituirle anche solo un centesimo di tutto il bene che mi dà.

Ringrazio Ludovica per tutte le chiacchierate fino all'alba discutendo di temi che mai avrei pensato di avere così tanto a cuore, per avermi fatto vivere Bologna e la mia casa come il posto perfetto per me.

Ringrazio Giada, che pur essendo stata la prima persona che ho conosciuto a Bologna, è tuttora quella con cui parlo per qualsiasi cosa importante accada nella mia vita.

Ringrazio Lorenzo, per avermi fatto scoprire l'importanza dell'informazione

e della passione per ciò che riteniamo parte di noi.
Ringrazio Umberto per essere diventato in così poco tempo una persona tanto importante per me, con cui è impossibile non passare giorni da ricordare.
Ringrazio Giorgio per tutte le storie raccontate e le serate in sala prove passate insieme, per le canzoni consigliate che ho ascoltato, per quelle che non ascolterò mai.
Ringrazio Giovanni, Luigi, Irene, Beatrice, Carmine, Francesca e numerosi Andrea, che sono sempre stati ciò che ha reso l' Università un periodo fantastico della mia vita.
Ringrazio chiunque, con qualsiasi gesto, mi abbia permesso di apprezzare le piccole cose.