Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

CONFRONTO TRA LIBRERIE E SERVIZI DI TRADUZIONE AUTOMATICA

Relatore:
Chiar.mo Prof.
Angelo Di Iorio

Presentata da:
Lorenzo De Blasiis

Sessione III
Anno Accademico 2022/2023

Indice

1 Introduzione

La traduzione automatica è un campo in evoluzione dell'intelligenza artificiale.
Esistono servizi online come Google Translate che da anni permettono di poter effettuare traduzioni in numerose lingue. Negli ultimi anni sono stati sviluppati altri servizi simili che danno la possibilità di essere utilizzati tramite API e ci sono alternative che permettono di fare traduzioni offline.
Il lavoro che segue ha lo scopo di rispondere a due domande:

1. Quali sono i servizi di traduzione automatica attualmente disponibili?
2. Qual è il miglior sistema di traduzione automatica, in particolar modo per la traduzione di notizie?

La prima parte del lavoro consiste in un'analisi su tutti i servizi che offrono traduzione automatica (machine translation) e il loro confronto in termini di funzionalità offerte e costo.
La seconda in un analisi sia qualitativa che quantitativa delle traduzioni fatte da questi servizi su un testbed composto da notizie.

1.1 TARO

L'esigenza di avere un confronto su diversi servizi di traduzione automatica nasce da TARO[4], un progetto nato per raccogliere e analizzare notizie di giornali online.
Lo scopo è quello di poter capire se la stessa notizia è stata pubblicata da giornali diversi, se viene ripubblicata in differenti formati e per quanto rimane online la stessa notizia.
Per raggiungere tale scopo servono dati quantitativi sul tempo di pubblicazione, per poter sapere quando una notizia è stata pubblicata e per quanto tempo è rimasta pubblicata.
Considerando giornali online di vari paesi le notizie pubblicate da un paese si diffonderanno in momenti diversi rispetto ad un altro anche in base fuso orario e ad altri fattori.
Il modello TARO deve distingue tra una notizia come concetto astratto e una notizia come oggetto concreto, la sua rappresentazione in forma testuale, video o audio.
Il modello TARO utilizza le seguenti definizioni:

1. Piece of News: l'entità astratta che rappresenta un evento che viene riportato. Un esempio potrebbe essere "Anniversario dell'allunaggio".

2. News item: un entità concreta che incarna un ”piece of news” e la rappresenta utilizzando il linguaggio naturale. Un esempio potrebbero essere differenti articoli di giornale che trattano la notizia ”Anniversario dell'allunaggio”.

La parte centrale dell'analisi di TARO consiste nel determinare se due ”news item” fanno riferimento alla stessa ”piece of news”.
Due ”news item” sono equivalenti se trattano della stessa notizia (”piece of news”), a prescindere dalla lingua, del media o del momento in cui è stato pubblicato.
Per effettuare le analisi sulle notizie vengono usate due metriche: la diffusione e la persistenza.
La diffusione indica quanti altri organi di stampa hanno pubblicato la stessa notizia (piece of news) mentre la persistenza quanto tempo una notizia rimane pubblicata. Entrambe vengono calcolate utilizzando degli snapshot, che verranno spiegati più nel dettaglio nel capitolo 3.1.
TARO può confrontare notizie provenienti da diverse fonti e in lingue diverse, ma l'analisi delle notizie deve essere effettuata tramite strumenti della libreria spaCy [1]in inglese. L'analisi semantica ha lo scopo di determinare quali notizie concrete appartengono alla stessa notizia astratta, serve quindi un metodo di traduzione affidabile per tradurre verso l'inglese da 4 lingue differenti: Italiano, Tedesco, Francese e Spagnolo. Migliore è la traduzione verso l'inglese migliore sarà anche la probabilità di collegare correttamente notizie concrete a notizie astratte.

Il servizio utilizzato nel articolo è Argos, una libreria open-source che può essere utilizzata offline scaricando il modello. Il lavoro che segue ha lo scopo di capire se sono presenti migliori soluzioni per la traduzione automatica di notizie, quali di queste sono disponibili gratuitamente e quali funzionalità offrono.

Nel prossimo capitolo prenderò in esame tutti i servizi di traduzione considerati per fare questa analisi.

[1]Sito ufficiale spaCy: https://spacy.io/

2 Servizi di traduzione automatica

La prima parte del lavoro è stata la ricerca di quali servizi sono presenti sul mercato, quali sono le loro caratteristiche, come si differenziano gli uni da gli altri e quali tecnologie utilizzano. Tutti i servizi utilizzano modelli creati con reti neurali NMT[1], pre-allenati con apprendimento supervisionato, tranne per openAI che non ha un modello alle traduzioni, ma utilizza un LLM (Large Language Model).
Al momento i modelli NMT sono lo standard per la traduzione automatica bilingua.

Precedentemente si utilizzava l'approccio statistico, abbreviato in SMT (Statistical Machine Translation)[3] introdotto da ricercatori IBM a inizi anni '90 [12]. Questo approccio si basa sulla distribuzione di probabilità che una data stringa α in una lingua di destinazione sia la traduzione di una stringa β nella lingua di origine. Il limite di questo metodo è che traduce frase per frase senza tenere conto del contesto.

L'attuale approccio abbreviato con la sigla NMT (Neural Machine Translation) si basa sull'utilizzo di reti neurali artificiali per generare le traduzioni. È in grado di catturare meglio il significato della frase e tradurre con una maggior precisione, ma le traduzioni risultano molto fedeli all'originale e troppo letterali.

Un ulteriore approccio, scoperto solo di recente, è quello di utilizzare un LLM dando come prompt il comando di tradurre dalla lingua di origine a quella di destinazione e fornire il testo da tradurre.
Sperimentando questo approccio su GPT3.5 ha dato come risultati una traduzione di alta qualità per le lingue su cui ha molte risorse, mentre capacità limitate per le lingue di cui ha poche risorse [7]. Si ha avuto risultati simili con GPT4 che è al pari dei migliori traduttori automatici per le traduzioni verso l'inglese, ma ha risultati peggiori con altre lingue poche rappresentate nel modello[11].

Servizi	Traslitterazione	Glossario	HTML	XML	Modelli personalizzati	Documenti	Language detection	Frasi di contesto	Offline	Lingue supportate
Deepl	Sì	Sì	Sì	Sì	No	Sì	Sì	No	No	31
Argos	No	No	Sì (plugin)	No	No	Si(plugin)	No	No	Sì	30
Google translation	Sì	Sì	Sì	No	No	No	Sì	No	No	100
Microsoft Translation	Sì	Sì	Sì	Sì	Sì	Sì	Sì	No	No	100
AWS Cloud Translation	Sì	Sì	Sì	No	No	No	Sì	Sì	No	75
Eden AI	No	No	No	No	No	Sì	Sì	No	No	262
IBM Cloud Translation	No	No	No	No	No	Sì	Sì	No	No	76

Figura 1: Tabella comparativa

La tabella comparativa di Figura 1 si riferisce alla versione gratuita per ogni servizio a pagamento: Microsoft tier S0, Google Translate Basic, Eden AI starter, IBM Cloud translation free, AWS 12 mesi gratuiti.
Il glossario, indicato nella tabella, è un insieme di parole o frasi con la rispettiva traduzione che il sistema di traduzione riconosce e sostituisce nell'output.
La "language detection" si riferisce alla possibilità di rilevare automaticamente la lingua del testo di input senza doverla specificare.
Mentre la colonna "Frasi di contesto" si riferisce alla possibilità di poter aggiungere ulteriore testo oltre a quello che si vuole tradurre allo scopo di migliorare la traduzione.

2.1 Argos Translate

Argos Translate[2] è una libreria open-source che può essere utilizzata offline, scritta in Pyton, che utilizza il modello pre-allenato di OpenNMT per tradurre testo in varie lingue, SentencePiece per la tokenization, Stanza per la Sentence boundary disambiguation.

OpenNMT[3] [8][9][10] è ecosistema opensource rilasciato con licenza MIT, sviluppato per machine translation e sequence learning che utilizza un'architettura di tipo Trasformer allenato con metodo di apprendimento supervisionato.

SentencePiece[4] è un tokenizer e detokenizer non supervisionato di testo, utile per i sistemi neurali che generano testo dove la dimensione del vocabolario è determinata prima del training del modello neurale.

Stanza[5] è una collezione di strumenti per l'analisi linguistica di molti linguaggi naturali. Partendo da semplice testo Stanza lo può dividere in frasi e parole, riconoscere le entità delle frase, fare analisi sintattica e altro.

Argos offre la scelta di molteplici interfacce per essere utilizzato: riga di comando, libreria python o con interfaccia grafica (GUI).
Data la natura open-source del progetto si sono sviluppate altre librerie e applicazioni attorno ad Argos come LibreTranslate[6], API e web-app che utilizza Argos come base, come HTML Library[7], libreria che permette di prendere in input e fare parsing di HTML, oppure come Argos Translate Files[8], libreria che permette di utilizzare Argos per tradurre file con le seguenti estensioni: `txt, odt, odp, docx, pptx, epub, html.`
È possibile richiedere di allenare modelli personalizzati su propri dati per 1000$ per ogni lingua.
Argos utilizza dei pacchetti con estensione ".argosmodel" che contengono i dati necessari per effettuare le traduzioni per ogni coppia di lingua in ingresso e in uscita. A

[2]GitHub: https://github.com/argosopentech/argos-translate,
sito del progetto: https://www.argosopentech.com/

[3]Sito ufficiale del progetto: https://opennmt.net/

[4]SentencePiece GitHub: https://github.com/google/sentencepiece

[5]Stanza GitHub: https://stanfordnlp.github.io/stanza/index.html

[6]Sito di LibreTranslate: https://libretranslate.com/,
GitHub LibreTranslate: https://github.com/LibreTranslate/LibreTranslate

[7]HTML library github: https://github.com/argosopentech/translate-html

[8]Argos Translate Files GitHub: https://github.com/LibreTranslate/argos-translate-files

questo indirizzo[9] si trovano tutti i pacchetti che possono essere scaricati. Se Argos non trova un pacchetto per la traduzione diretta da un lingua a un'altra effettua automaticamente un traduzione a una lingua intermedia e poi dalla lingua intermedia a quella in output richiesta. In questo modo se voglio per esempio effettuare una traduzione dal russo allo spagnolo, posso farlo anche in assenza del pacchetto di traduzione da russo a spagnolo perché ho installato il pacchetto che traduce da russo a inglese e quello che traduce da inglese a spagnolo, quindi invece di tradurre russo → spagnolo, utilizzando l'inglese come lingua intermedia: russo → inglese → spagnolo. Si perde qualità nella traduzione, ma permette di tradurre un maggior numero di lingue.
Le lingue supportate sono 30 e si possono trovare sulla pagina Github [1] del progetto. È in fase di sviluppo una versione 2 in beta, che utilizza un'architettura multilingua, questo permetterà di evitare di passare dall'inglese come lingua intermedia. Superando il limite della prima versione che fornisce solo pacchetti di traduzione da o verso l'inglese. Purtroppo non sono stato in grado di utilizzarla perché non sono disponibili i pacchetti di traduzione per la nuova versione e per la mancanza di documentazione per utilizzare correttamente il modello.

2.2 Deepl API

Deepl è un sito che offre un'interfaccia web[10] per tradurre testo da una lingua a un'altra in modo simile a Google Translate, presenta anche una modalità per tradurre interi documenti.
Oltre a questo offre un API[11], gratuita fino a 500.000 caratteri al mese. Le API possono essere utilizzare tramite cURL o equivalenti, ma per facilitare le interrogazioni vengono fornite librerie per i principali linguaggi di programmazione: .NET, PHP, Node.js, Python, Ruby, Java.
L'API fornisce varie opzioni per modificare l'output e di seguito le esaminerò una per una:

1. Contesto: è possibile includere testo, che serve come contesto, per influenzare la traduzione, ma senza che venga tradotto. Questo può essere utile per migliorare la qualità della traduzione specialmente con frasi molto brevi. Questa funzionalità è ancora in fase alpha di cui non si consiglia l'utilizzo in produzione.

2. Divisione in frasi: questa opzione, attivata di default, divide il testo in frasi tramite i punti utilizzando la punteggiatura (”.” e ”;”), traduce le frasi e ricompone

[9] https://www.argosopentech.com/argospm/index/
[10] DeepL web-app: https://www.deepl.com/translator
[11] DeepL API: https://www.deepl.com/pro-api?cta=header-pro-api

il testo. Dato che le API di DeepL non possono tradurre frasi estremamente lunghe, se si disattiva la divisione in frasi automatica le frasi troppo lunghe potrebbero essere terminate prima della loro fine, in questo caso è consigliato dividere le frasi manualmente prima di effettuare la richiesta.

3. Preservare la formattazione: preserva la formattazione originale del testo in input. Normalmente il motore di traduzione corregge alcuni aspetti della formattazione come la punteggiatura alla fine di una frase, o la lettera maiuscola all'inizio di una frase.

4. Formalità: con 4 opzioni di cui una di default, permette di scegliere il grado di formalità del testo in output. Questa opzione è disponibile solo per le seguenti lingue: Italiano, Tedesco, Francese, Spagnolo, Olandese, Polacco, Portoghese, Russo.

5. Tag handling: permette di selezionare quali tipi di tag gestire se i tag XML o i tag HTML

6. Glossario: Si possono creare glossari per ogni coppia di lingue. Si può creare un glossario specificando lingua sorgente, lingua di destinazione e una lista `csv` o `tsv` di termini. Questi termini verranno tradotti utilizzando il glossario invece del motore di traduzione di DeepL

7. Outline manuale: il parametro `outline_detection` disattiva la rilevazione automatica dei tag quando il tag handling è attivo. Si può utilizzare l'opzione `splitting_tags` e `non_splitting_tags` per decidere quali tag costituiscono la struttura e dividono le frasi che devono essere tradotte e quali tag non dividono mai le frasi. Si può usare anche `ignore_tags` per indicare tag che contengono testo che non deve essere tradotto. Con queste impostazioni si ha un maggiore controllo su come deve essere tradotto un file XML o HTML.

È possibile tradurre grandi quantità di testo inserendo interi paragrafi nel parametro di testo della richiesta. Per ogni richiesta si possono inserire fino a 50 parametri da tradurre facendo più chiamate in parallelo.
Se nel testo input sono presenti dei caratteri speciali o sequenze di caratteri non comuni, magari necessari per un linguaggio di markup, potrebbero venire modificati, tradotti o eliminati, bisogna quindi fare attenzione e utilizzare le opzioni discusse sopra o dividere già le frasi in maniera da evitare di includere nel testo dei marker.
DeepL può anche tradurre documenti nei formati `docx`, `pptx`, `xlsx`, `pdf`, `htm/html`, `txt`, `xlf/xliff`. Per ogni documento tradotto nei formati `docx`, `pptx`, `xlsx`, `pdf` vengono utilizzati un minimo di 50,000 caratteri.

Le opzioni per la traduzione di documenti comprendono il glossario e la formalità e funzionano esattamente come è stato descritto precedentemente.
Vi è la possibilità di scegliere il formato del file in output, se non viene specificato verrà usato lo stesso del file di input, ma funziona unicamente da formato `pdf` a formato `docx`.
DeepL supporta 31 lingue, la lista completa si trova nella documentazione ufficiale delle API [12].
Maggiore è il contesto fornito migliore è la traduzione, questo può avvenire utilizzando il parametro del contesto o semplicemente mettendo più testo rilevante all'interno di una singola richiesta.
I vari piani di pagamento forniscono le medesime features, ma con limiti di utilizzo diverso.
Il dataset su cui è stato allenato inizialmente il modello viene preso da "Linguee"[13], un dizionario online che fornisce traduzioni ed esempi di frasi tradotte. Successivamente il dataset è stato ampliato utilizzando un web-crawler.
Il modello di Deepl ha un'architettura Trasformer con una metodologia di apprendimento supervisionato, ulteriori dettagli vengono tenuti nascosti.

2.3 Google Cloud Translate API

Google Cloud Translation[14] offre un modello pre-allenato basato su reti neurali. Offre un API con 500.000 caratteri gratuiti al mese, 300$ di crediti per poter provare altre funzionalità più avanzate e librerie per i principali linguaggi di programmazione. Google Translate supporta la traduzione di più di 100 lingue[14].
La versione base gratuita mette a disposizione un modello pre-allenato NMT (Google Neural Machine Translation) ottimizzato per la semplicità e scalabilità.
Può prendere come input HTML, ma non XML o altri linguaggi di markup, mentre può rilevare automaticamente quale è la lingua utilizzata nell'input.
La versione avanzata offre le stesse funzionalità della versione base e in più:

1. Possibilità di allenare modelli personalizzati per tradurre testo che contiene parole di un contesto specifico

2. Traduzione batch con Cloud Storage: si può prendere l'output da Cloud Storage e fare una richiesta asincrona di traduzione, quando l'operazione è stata completata

[12]Lista di lingue supportate da DeepL: https://developers.deepl.com/docs/resources/supported-languages
[13]Linguee: https://www.linguee.it/
[14]Lista completa lingue supportate da Google Translate API: https://cloud.google.com/translate/docs/languages

l'ouput viene scritto su un bucket di Cloud Storage specificato.

3. Traduzione di documenti, i formati supportati sono: `doc`, `docx`, `pdf`, `ppt`, `pptx`, `xls`, `xlsx`

4. Glossario: insieme di coppie di parole per ogni lingua in un formato .tmx, .csv, .tsv

5. Traduzione con modelli diversi da NMT, un esempio è AutoML che può essere personalizzato fornendo dati di un dominio del linguaggio specifico come ad esempio documenti tecnici. Rendendolo più adatto a tradurre quel tipo di linguaggio.

6. Supporto alla romanizzazione: se l'output è in una lingua che non utilizza caratteri latini (per esempio il giapponese) viene convertito in caratteri latini basandosi sulla pronuncia.

7. Supporto alla traslitterazione: se l'output è in una lingua che non utilizza caratteri latini (per esempio il giapponese) viene traslitterato, ovvero vengono mappati i simboli da un alfabeto a un altro che hanno una somiglianza fonetica.

8. Traduzione adattiva: insieme al testo da tradurre si fornisce anche un esempio di traduzione che verrà utilizzato per migliorare la traduzione. Con questo metodo bisogna utilizzare un modello LLM (Large Language Model).

Ci sono delle librerie per facilitare l'utilizzo delle API per i seguenti linguaggi di programmazione: GO, Python, Java, Node.js, C#, PHP, Ruby.
Il costo della traduzione con il modello NMT è di 20$ per ogni milione di caratteri tradotti dopo che si superano i primi 500.000 caratteri al mese. Per i modelli personalizzati il costo è sempre per ogni milione di caratteri tradotti e diminuisce se vengono tradotti più caratteri in certe soglie.

2.4 Amazon Translate API

Amazon Translate API è uno dei servizi compresi con AWS di cui esiste una versione di prova gratuita per 12 mesi,in cui è possibile tradurre fino a 2 milioni di caratteri al mese.
Si può accedere alle API tramite gli AWS SDK, la linea di comandi di AWS o dalla console di AWS con un'interfaccia grafica. Gli SDK sono disponibili per vari linguaggi di programmazione come: C++, Python, Java, GO, Javascript, Kotlin, .NET, Node.js, PHP, Ruby, Swift, Rust, SAP ABAP.

Supporta la traduzione di 75 lingue[15].
Amazon invece di offrire la possibilità di creare il proprio modello personalizzato allenandolo con i propri dati, utilizza un sistema che chiamano "Active Custom Translation (ACT)" in cui bisogna fornire dei dati di traduzione chiamati "Parallel Data", consistono in esempi che mostrano come si vuole che il testo venga tradotto, per ogni esempio ci deve essere la lingua di origine e tutte le lingue per cui si vuole ottenere una traduzione. ACT userà queste informazioni per influenzare la traduzione. Migliorando la traduzione in domini specifici del linguaggio senza dover allenare un nuovo modello e senza modificare la qualità delle traduzioni per i casi non specifici per cui il modello è già stato allenato.
Come altri servizi, visti precedentemente, Amazon translate permette di utilizzare un glossario in un formato `csv`, `tsv`, o `tmx`, può rilevare automaticamente quale lingua si sta utilizzando come input, può tradurre documenti nei formati HTML e Docx e permette di fare traduzioni di più testi o documenti in una sola richiesta (batch).
Inoltre offre la funzionalità di glossario in cui un insieme di frasi già tradotte può essere sostituito a specifiche frasi nell'output.
L'API ha un limite di 10.000 byte per richiesta, il costo del servizio è di 15$ per ogni milione di caratteri tradotti sia per il testo che per i documenti. Mentre per le traduzioni fatte con ACT, 60$ per ogni milione di caratteri.

2.5 Microsoft Text Translation API

Microsoft offre un API con cui si può tradurre fino a 2 milioni di caratteri al mese gratuitamente.
Viene fornito un SDK per i seguenti linguaggi: Python, C#, Java, Javascript.
Nella versione gratuita sono disponibili le seguenti funzionalità:

1. Rilevazione automatica della lingua

2. Glossario: come visto in precedenza serve per tradurre una lista di parole con una corrispondente traduzione ogni volta che tali parole vengono tradotte, vi è inoltre la possibilità di avere un glossario di frasi che agisce come una funzione cerca e sostituisci.
   Oppure può essere utilizzato un dizionario dinamico che consiste nel fornire la traduzione di una parola o frase direttamente nell'input della richiesta tramite un markup.

[15]Lista completa delle lingue supportate da AWS:
https://docs.aws.amazon.com/translate/latest/dg/what-is-languages.html

Esempio:

```
        Input(EN): The word <mstrans:dictionary translation="wordomatic">
wordomatic</mstrans:dictionary> is a dictionary entry.
        Output(DE): Das Wort "wordomatic" ist ein Wörterbucheintrag
```

3. Dizionari neurali: invece di agire come semplici cerca-e-sostituisci usano le parole e le frasi del glossario e le modificano adattandole al contesto, possono per esempio modificare maiuscole e minuscole o cambiare il genere di una parola. È una funzionalità disponibile solo per un numero limitato di coppie di lingue.

4. Traslitterazione

5. Allenamento di modelli personalizzati

Con la versione a pagamento si aggiunge la possibilità di tradurre documenti e la funzionalità "Parallel Documents", grazie a cui si possono personalizzare le traduzioni fornendo documenti già tradotti su terminologia specifica.
I formati accettati sono `TMX`, `XLIFF`, `TXT`, `DOCX`, `XLSX`.
Per usare questa funzionalità bisogna fornire una coppia di documenti con almeno 10.000 frasi presenti in entrambi i documenti e allineate, ovvero ci deve essere una corrispondenza tra ogni frase e la sua traduzione. In aggiunta è possibile caricare altri documenti che contengono le stesse frasi, ma tradotte in un'altra lingua e il sistema le metterà in corrispondenza con quelle degli altri documenti. È possibile inserire altri dati in una sola lingua per migliorare la traduzione della lingua stessa. In questo modo si può allenare un modello personalizzato.
Sono supportate più di 100 lingue[16].
Si può creare un nuovo modello utilizzando soltanto i glossari, in questa modalità non c'è un limite minimo di frasi da dover utilizzare, in questo modo verrà allenato più velocemente rispetto al metodo tradizionale.
Il costo è di circa 9€ per milione di caratteri tradotti.

2.6 Eden AI API

Eden AI[17] è un servizio pensato per le imprese per offrire in un'unica interfaccia API vari servizi di intelligenza artificiale di più fornitori, tra i quali molti di quelli menzionati precedentemente.

[16]Lista delle lingue supportate da Microsoft Text Translation:
https://www.microsoft.com/en-us/translator/languages/
[17]Sito ufficiale di EdenAI: https://www.edenai.co/

I fornitori che si possono scegliere sono: Amazon, Google, IBM, Microsoft, Neuralspace, ModernMT, Phedone, OpenAI.
Il servizio offre 10$ di crediti per poterlo provare così l'ho utilizzato per accedere ad alcuni fornitori che non hanno un piano gratuito, ovvero OpenAI e ModernMT.
Per fare una richiesta di traduzione è necessario fornire una serie di parametri utili alla traduzione e una lista dei fornitori da utilizzare, sarà la stessa API ad eseguire le richieste specifiche per ogni fornitore. Risulta così piuttosto opaco il preciso meccanismo di traduzione che viene effettivamente utilizzato. Deduco riguardo a OpenAI che il testo venga dato in input a GPT3.5 o GPT4 preceduto dal prompt: "Traduci il seguente testo da [lingua input] a [lingua output]", infatti in alcuni casi di test nell'output era presente un prompt simile prima del testo tradotto.

ModernMT[2][5][6] è un servizio di machine translation basato sul modello Fairseq Transformer, uno dei loro servizi a pagamento è quello di poter migliorare le traduzioni automatiche con un basso punteggio di confidenza tramite dei traduttori umani senza l'intervento dell'utente, con un costo variabile a seconda della quantità di revisioni fatte dai traduttori, chiamano questo sistema "Human in the Loop".
È stato annunciato, per i clienti della versione a pagamento, la possibilità di allenare modelli personalizzati sui propri dati.

EdenAI supporta la traduzione in 262 lingue[18].
Nella versione gratuita c'è un limite di 5 chiamate alle API al minuto che rende molto lento il processo di traduzione di più input. Con i piani a pagamento i limiti sono più alti, con il piano più costoso a 500$ al mese si arriva ad avere un massimo di 1000 chiamate al minuto che comunque limita ed impedisce la scalabilità di determinate soluzioni.

2.7 IBM Cloud Translation (Ex Watson)

IBM Cloud Translation, noto anche come Watson® Language Translator è attualmente deprecato, a giugno 2024 il servizio non verrà più supportato e il 10 dicembre 2024 sarà ritirato e non sarà più disponibile.
Il sistema di IBM è un modello NMT pre-allenato sviluppato usando reti neurali di tipo Trasformer.
Offre un API gratuita che permette di tradurre fino a un milione di caratteri al mese con un modello di traduzione di default e permette la traduzione di documenti nei

[18]Lista completa delle lingue supportate da EdenAI:
https://docs.edenai.co/reference/translation_automatic_translation_create

formati `json`, `xml`, `html`, `pdf`, `word`, `powerpoint`. Supporta la traduzione di 76 lingue [19].
Con piani a pagamento si possono tradurre un maggiore numero di caratteri al mese, tradurre documenti di dimensioni maggiori e utilizzare dei modelli di traduzione personalizzati per termini specifici di un certo settore.

2.8 Servizi esclusi

Di seguito elenco i servizi che non sono stati utilizzati poiché presentano problemi per il mio caso di studio. Essi sono:

1. Phedone[20]: servizio di NLP (Natural Language Processing) creato da un'azienda francese.
   Ho scelto di non utilizzare questo servizio dal sito ufficiale perché la versione gratuita di prova offre solo 100 chiamate, troppo poche per il mio utilizzo.
   È possibile utilizzarlo attraverso Eden AI, ma in questo caso l'unita minima di fatturazione è 1000 caratteri e poiché nel mio caso studio prendo in esame testi con meno di 1000 caratteri ho scelto altri servizi per allocare meglio i crediti che avevo a disposizione.

2. Neuralspace[21]: Offre vari servizi di NLP con intelligenza artificiale per l'analisi di audio, documenti e testo, tra le funzioni per l'analisi del testo vi è anche la traduzione. Supporta 148 lingue, includendo molte lingue non presenti in altri modelli. [22].
   Esiste un piano gratuito con 100$ di crediti per poter provare il servizio, iscrivendosi alla piattaforma e ogni richiesta costa circa 0.001$.
   Nonostante io mi sia iscritto e abbia ricevuto la mail con il link per attivare l'account mi è stato impossibile procedere con l'attivazione dell'account.
   Avrei potuto usare questo servizio tramite Eden AI, ma l'unita minima di fatturazione era una richiesta, nuovamente non adeguata per il mio utilizzo poiché avrebbe aumentato troppo il costo limitandomi nell'utilizzo di altri servizi.

3. MBart[23]: è un modello pre-allenato su testi monolinguistici in 25 lingue, open-source e sviluppato da Meta.

[19] Lista completa delle lingue supportate da IBM:
https://www.ibm.com/docs/en/watson-libraries?topic=references-supported-languages
[20] Sito ufficiale Phedone: https://phedone.com/en/
[21] Sito ufficiale di NeuralSpace: https://docs.neuralspace.ai/machine-translation/overview
[22] Lista completa delle lingue supportate:
https://docs.neuralspace.ai/machine-translation/language-support
[23] Github di MBart: https://github.com/facebookresearch/fairseq/blob/main/examples/mbart/README.md

È progettato per essere utilizzato come base per fare fine-tuning, ma non avendo trovato sufficienti documentazioni non sono riuscito ad utilizzarlo per fare traduzioni.

4. T5[24]: È un modello pre-allenato su più task sia con apprendimento supervisionato che con apprendimento non supervisionato. E' stato utilizzato come dataset il "Colossal Clean Crawled Corpus" (C4), rilasciato nello stesso articolo in cui è stato rilasciato T5 [13].
   Progettato per eseguire diversi compiti text-to-text tra cui la traduzione. Mostra il codice necessario per utilizzare il task di traduzione, ma non funziona come previsto poichè modificando l'input per ottenere traduzioni in altre lingue il testo non viene più tradotto o si ottengono altri comportamenti non voluti.

5. HuggingFace pipeline[25]: la piattaforma di condivisione di modelli di intelligenza artificiale HuggingFace fornisce una libreria Python chiamata pipeline.
   Le pipeline di HuggingFace sono oggetti che semplificano l'uso dei modelli per l'inferenza offrendo un'API semplice dedicata a vari compiti tra cui anche la traduzione.
   L'utilizzo è piuttosto semplice, basta dare in input il modello che si vuole utilizzare che verrà scaricato automaticamente da HuggingFace e poi il prompt per generare testo.
   Ho provato vari modelli disponibili per traduzioni ottenendo delle traduzioni corrette, ma cambiando lingua di origine e lingua di destinazione la traduzione non avveniva più correttamente. Dato che la traduzione funzionava solo per alcune delle coppie di lingue che dovevo tradurre ho escluso l'utilizzo di questo servizio.

Nel prossimo capitolo spiegherò in dettaglio il metodo utilizzato per valutare questi servizi.

[24]Link Huggingface T5: https://huggingface.co/google-t5/t5-base

[25]Huggingface pipeline:
https://huggingface.co/docs/transformers/v4.35.2/en/main_classes/pipelines#transformers.pipeline

3 Valutazione Servizi di traduzione

Descrivo brevemente il metodo di lavoro in generale per poi andare nello specifico nelle sezioni seguenti.
Il primo passaggio è stato quello di creare dei testbed, usati come input dei servizi di traduzione.
Sono presenti in totale quattro testbed, dei quali tre sono stati creati con le notizie prese dagli snapshot di TARO, utilizzando diverse categorie di argomenti, mentre il quarto è stato creato da un documento dell'Unione Europea (Acquis Communautaire), contenente già le traduzioni in inglese. Quest'ultimo è stato utilizzato come golden standard per poter fare il confronto tra il testo generato dalla traduzione automatica e quello creato da traduttori professionisti.
Per ognuno di questi testbed creo un documento con le traduzioni verso l'inglese per ognuno degli 8 servizi elencati nel capitolo 2, conteggiando Eden AI due volte sia come OpenAI che come ModernMT.

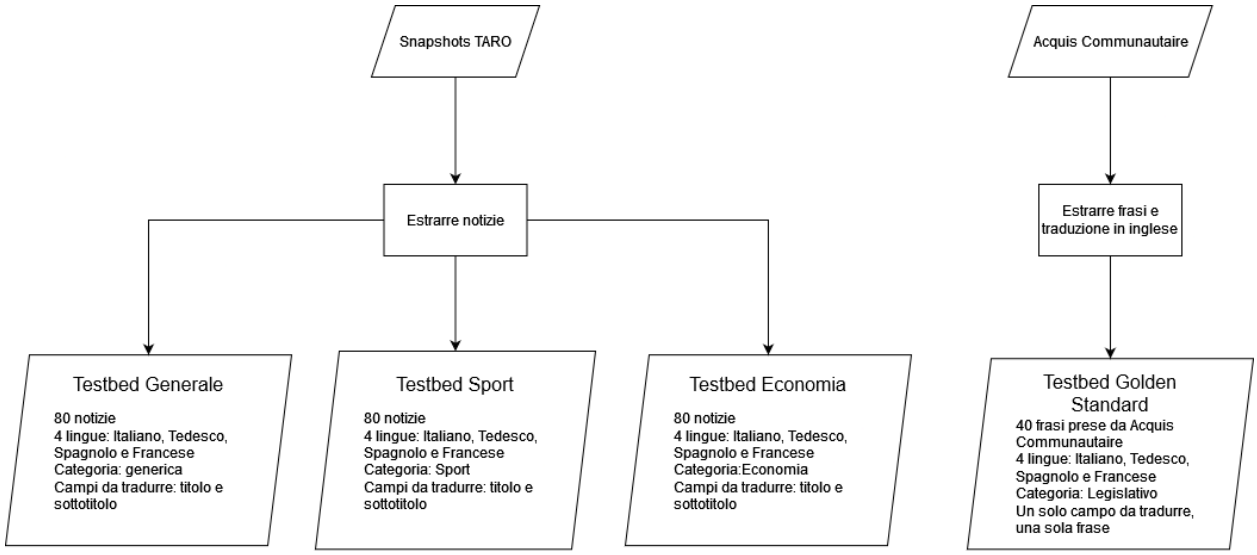

Figura 2: Testbed

Su questo documento effettuo un'analisi in due passaggi, prima qualitativa e poi quantitativa.
La prima serve innanzitutto per controllare che i servizi abbiano restituito l'output desiderato, in secondo luogo che le frasi inglesi siano frasi corrette e di senso compiuto. Procedo quindi con l'analisi quantitativa dove calcolo quanto sono simili i testi tradotti tra i vari servizi utilizzando vari algoritmi che calcolano la similarità, dando

un punteggio che va da 0, per frasi completamente diverse, a 1, per frasi completamente uguali.
Sono state utilizzate le notizie raccolte tramite gli snapshot di TARO per creare 3 testbed in ambiti diversi.

Le lingue prese in esame sono Italiano, Francese, Tedesco e Spagnolo.
Di seguito riporto come esempio una notizia codificata in TARO, ho eliminato una parte del "content" per evitare di occupare troppo spazio.

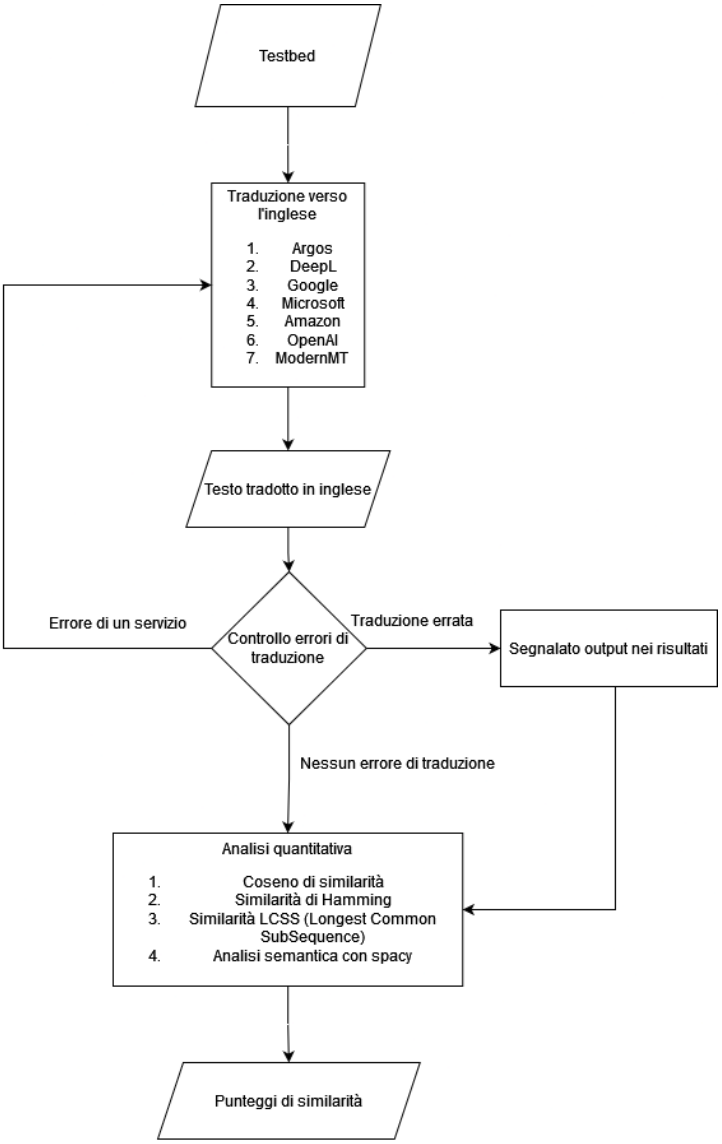

Figura 3: Diagramma analisi

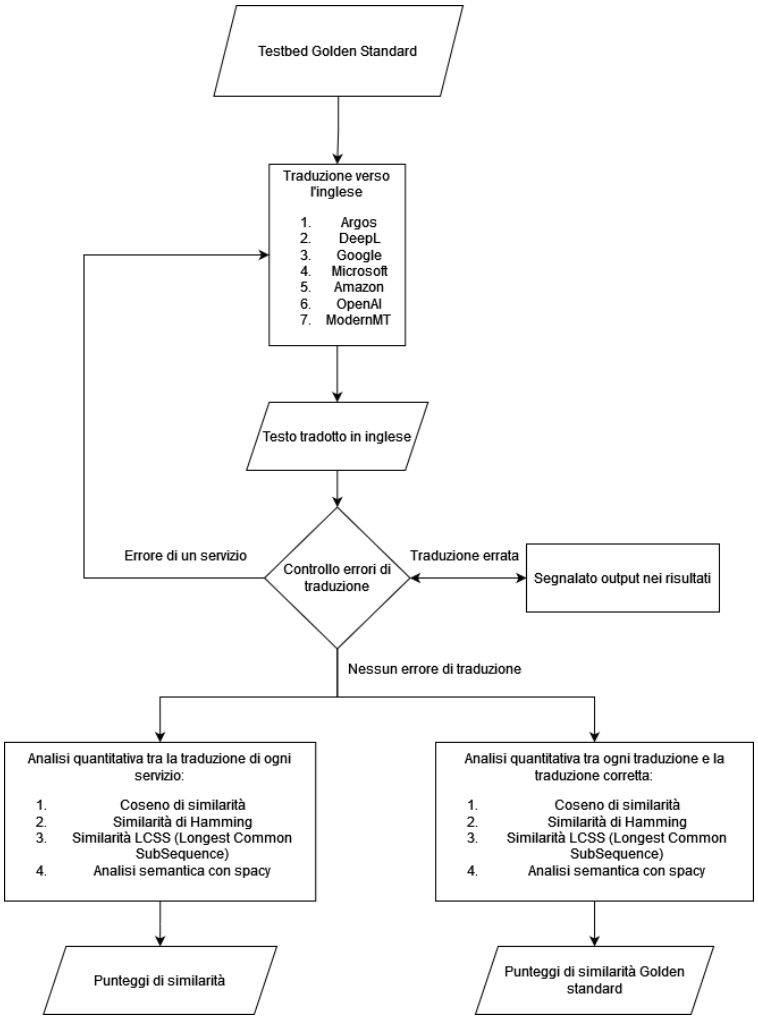

Figura 4: Diagramma analisi Golden Standard

3.1 Raccolta dei dati

TARO utilizza gli snapshot per poter fare analisi quantitative su flussi di notizie come le pagine web di giornali online. Uno snapshot in TARO è un insieme di notizie pubblicate da un organo di stampa in uno specifico momento del tempo.
Per i media in cui vengono pubblicate costantemente notizie senza avere un'edizione, come avviene invece per i giornali o telegiornali, si crea uno snapshot delle notizie di un flusso di notizie. Lo snapshot è formato da un insieme di news item pubblicati in un dato momento.Un esempio è "Il feed RSS di PBS News, apparso online il 14/03/2023 alle 17:15:00 UTC".
Le proprietà di uno snapshot sono:

1. outlet: l'organo di stampa che ha pubblicato i news item contenuti nello snapshot

2. data di pubblicazione

3. argomento: una descrizione degli argomenti comuni a tutte le notizie dello snapshot

Esempio di uno snapshot:


```
      {
  "title": "Quanto è dipendente la Germania dal gas russo",
  "date_raw": "April 05, 2022",
  "date": "2022-04-05",
  "url": "https://www.ilpost.it/economia/",
  "news_url": "https://www.ilpost.it/2022/04/05/germania-gas-russo/",
  "subtitle": " Tantissimo: ed è il motivo per cui in questi giorni si sta opponendo a
  nuove sanzioni europee che riguardino il gas ",
  "content": "In questi giorni i governi e le istituzioni europee stanno discutendo un
  nuovo pacchetto di sanzioni da approvare contro la Russia per via 'dellinvasione
  'dellUcraina. Sarà il quinto pacchetto di sanzioni 'dallinizio della guerra, ma
  difficilmente sarà il più duro.Secondo diversi osservatori e analisti, ormai da
  settimane la misura più efficace per colpire 'leconomia russa nel brevissimo termine
  sarebbe un divieto delle importazioni di gas naturale russo 'nellUnione Europea, che
  nel 2020  il 38,1 per cento del gas naturale importato 'dallesteso proprio dalla
  Russia. Ma una misura del genere non è nemmeno in discussione. La Germania, il paese
  più ricco e influente 'nellUnione Europea, ha già fatto sapere che si opporrà a
  qualsiasi riduzione delle forniture di gas russo, da cui dipendono enormi pezzi della
  sua economia.[...]",
  "ranked": 0,
  "placed": "Economics",
  "epoch": 1649251811.163369,
  "language": "IT",
  "source": "ilPost"
}
```


Il primo testbed è composto da 20 notizie, di carattere generale, per ogni lingua. Sono stati creati altri due testbed, sempre con 20 notizie per ogni lingua, prendendo solo notizie di sezioni specifiche delle testate giornalistiche: Sport ed Economia. Sono state scelte queste due sezioni per osservare come cambia la qualità delle traduzioni rispetto al caso generale perché in questi due ambiti si utilizzano termini specifici, giochi di parole e locuzioni particolari per creare i titoli delle notizie. La divisione in categorie delle notizie è già stata fatta da TARO e mi sono affidato a quella per scegliere le notizie di Sport ed Economia.
Le notizie di carattere generale sono state prese nello stesso giorno e se possibile dallo snapshot della stessa ora del giorno. Ogni lingua è stata presa da un unico snapshot, quello delle 15:30 del 06-04-2022.
Le notizie di settore non è stato possibile prenderle dallo stesso snapshot o da snapshot di un'unica giornata, poichè troppo poche e spesso si ripetevano anche a distanza di più giorni.

Infine ho creato un file `json` per ogni testbed con le 40 notizie.
Le notizie sono composte da da 3 campi testo: titolo, sottotitolo e contenuto.
Inizialmente ho fatto la traduzione solo dei titoli per verificare che il sistema di traduzioni funzionasse, successivamente ho espanso la traduzione ai sottotitoli, mentre il contenuto ho deciso di non tradurlo per non superare il limite gratuito di caratteri di alcuni servizi.
Ogni notizia contiene un campo con la lingua della notizia in codice ISO 639 (Italiano → IT, Tedesco → DE, Spagnolo → ES, Francese → FR). Questo codice serve per essere passato come input insieme al testo da tradurre per i servizi che non hanno la rilevazione automatica della lingua.

3.2 Traduzione

Prima di effettuare le traduzioni è necessario preparare l'input composto da titolo e sottotitolo.
Alcuni sottotitoli delle notizie in spagnolo hanno una copia del contenuto, invece di avere un sottotitolo, risulta così troppo lungo rispetto ai sottotitoli delle altre lingue e porta a modificare i risultati delle analisi e consumare troppi caratteri. Per risolvere il problema ho limitato tutti i sottotitoli a una lunghezza massima.
Per evitare di tagliare le frasi o le parole prima della loro conclusione ho utilizzato spaCy per analizzare le frasi e limitare il sottotitolo alla prima frase.
Per cercare di migliorare i risultati delle traduzioni ho provato a concatenare il sottotitolo al titolo e inviarlo in una singola richiesta in modo da avere più contesto per il sistema di traduzione. Non sempre questo è andato a buon fine e per poter dividere il titolo e il sottotitolo tradotto nell'output ho dovuto utilizzare dei caratteri speciali raramente utilizzati in un testo. In rari casi questi caratteri divisori vengono convertiti in altri caratteri o eliminati, in questi casi traduco titolo e sottotitolo in due richieste separate.

3.3 Analisi della similarità

Dopo aver ottenuto la traduzione di titoli e sottotitoli dagli 8 provider per ogni notizia ho fatto un primo controllo manuale per segnalare tutte le traduzioni non corrette, questi casi verranno discussi nella sezione 4.1.
Successivamente ho effettuato un'analisi quantitativa dove in primis calcolo la similarità tra tutte le traduzioni utilizzando tre algoritmi di similarità : coseno, hamming e LCSS (Longest common subsequence). La libreria che ho utilizzato che implementa questi algoritmi è textdistance[26].

[26]Libreria textdistance: https://pypi.org/project/textdistance/

La similarità nell'analisi testuale è una tecnica che misura quanto due testi sono simili tra loro, ogni algoritmo calcola questa similarità in modo diverso.
Sono stati scelti quattro algoritmi diversi che agiscono con logiche diverse.

Il coseno di similarità[27] è una tecnica euristica che calcola il coseno tra due vettori, che contengono la frequenza dei termini. Il risultato è un numero compreso tra 0, nessuna parola è presente in entrambi i testi, e 1, dove i termini contenuti nei due testi sono gli stessi non necessariamente nello stesso ordine.
Questo algoritmo non tiene conto della posizione delle parole all'interno della frase COSì ho deciso di utilizzare anche un altro algoritmo in cui frasi con parole in ordine diverso non vengono considerate uguali, come l'algoritmo della distanza di Hamming.

L'algoritmo di Hamming[28] misura il minimo numero di sostituzioni necessarie per trasformare una stringa in un' altra. A differenza del coseno di similarità non si basa sulle parole, ma sui singoli caratteri.
Per confrontarlo con il coseno di similarità e con gli altri algoritmi ho considerato la normalizzazione della similarità di Hamming, dando valore 0 per massima distanza di Hamming cioè testi differenti, e valore 1 per testi completamente uguali carattere per carattere.

Oltre a questi due ho utilizzato anche l'algoritmo LCSS che segue principi diversi per avere un migliore confronto con gli altri.
LCSS, acronimo per Longest Common SubSequence[29], calcola la sottosequenza più lunga comune ai due testi considerando che una sequenza non è composta da caratteri consecutivi. Questo algoritmo osserva la struttura della frase e quindi è importante la posizione delle parole.
Anche in questo caso ho utilizzato una normalizzazione, per ottenere un range da 0 a 1 e poter essere così confrontato con gli altri.

Verso la fine del caso studio ho deciso di affiancare a questi algoritmi di analisi testuale un'analisi semantica tramite la libreria spaCy[30] perché analizzando i risultati mi

[27]Pagina wikipedia su coseno di similiarità:
https://it.wikipedia.org/wiki/Coseno_di_similitudine?oldformat=true
[28]Pagina Wikipedia sulla distanza di Hamming:
https://it.wikipedia.org/wiki/Distanza_di_Hamming?oldformat=true
[29]Pagina Wikipedia su LCSS:
https://it.wikipedia.org/wiki/Massima_sottosequenza_comune?oldformat=true
[30]Documentazione di spaCy sulla similarità semantica:
https://spacy.io/usage/linguistic-features#vectors-similarity

sono accorto che nella quasi totalità dei casi le traduzioni differivano più sulla forma che sul contenuto: usando le stesse parole, ma in ordine diverso, dando punteggi alti con la similarità del coseno e LCSS, ma bassa con Hamming. Tuttavia il contenuto della frase rimaneva il medesimo e cambiava solo la sua forma.
La similarità di spaCy dà come risultato 1, cioè testi con lo stesso significato, se composti dalle stesse parole in posizioni diverse, poiché l'analisi viene principalmente fatta a livello del significato delle parole.
Tutti questi risultati vengono scritti per ogni coppia di servizi e per ogni notizia in un file `json`. Questo procedimento avviene per ogni testbed e quindi avrò un output con i punteggi di similarità per ogni testbed.
Il testbed Golden standard è stato utilizzato anche in questo caso, ma senza tenere conto delle frasi già tradotte per poter avere un ulteriore caso di studio su frasi con un linguaggio tecnico. Dati i punteggi di similarità calcolo una media di ogni algoritmo.

Ho prodotto diverse tipologie di grafici per mostrare sotto vari punti di vista il comportamento di tutti i servizi.
Il grafico di tipo heatmap non è altro che una tabella a doppia entrata con la caratteristica che sia sulle righe che sulle colonne ci sono gli stessi servizi. Per ogni coppia di servizi c'è una casella con un valore calcolato facendo la media dei coseni di similarità di quella coppia di servizi per tutte le notizie del testbed.
Gli stessi dati sulle medie vengono anche rappresentati in un istogramma con due colonne, una per i titoli e una per i sottotitoli per mostrare le differenze tra i due tipi di testi tradotti.
Ho creato un grafico di tipo boxplot che rappresenta le mediane, per capire al meglio come leggere questo grafico spiego brevemente come è composto.
La parte centrale del grafico è rappresentata dal rettangolo colorato che chiamerò scatola la cui linea centrale è la mediana che indica che il 50% dei dati ha valori inferiori a quello indicato.
La linea superiore mostra il terzo quartile, cioè il 75% dei dati è inferiore a questo valore, mentre la linea inferiore rappresenta il primo quartile, cioè il 25% dei dati è inferiore a questo valore.
Le due linee che estendono la scatola vengono chiamati "whiskers" o baffi ed hanno una lunghezza di 1,5 volte l'intervallo interquartile ovvero la distanza tra il terzo quartile e il primo quartile, l'altezza della scatola. Tutti i singoli dati che sono al di fuori dell'intervallo dei baffi sono considerati anomalie o "outliers" e vengono rappresentati da cerchi.
Infine è presente una altro istogramma creato calcolando le medie dei punteggi di similarità senza fare distinzione tra titoli e sottotitoli per i quattro algoritmi scelti.

3.4 Confronti fatti con frasi legislative

Nella sezione precedente ho confrontato i risultati delle traduzioni tra di loro, ma non c'era la possibilità di confrontarlo con una traduzione corretta fatta da un traduttore umano. Qui invece confronto le traduzioni fatte dai servizi di traduzione con quelle fatte da traduttori professionisti. Il materiale è stato preso da un dataset[31] formato da segmenti del corpus legislativo dell'Unione Europea Acquis Communautaire. Contiene un insieme di leggi, trattati, regolamenti e direttive adottate dall'Unione Europea. Questo corpus è stato tradotto in tutte e 24 le lingue ufficiali dell'UE.

Su queste traduzioni ho calcolato gli stessi punteggi di similarità calcolati sui testbed precedenti, ma invece che calcolarli tra le diverse traduzioni sono il risultato del confronto tra la traduzione presente in Acquis e quelle fatte da vari servizi ottenendo così un punteggio che rappresenta la similarità tra il testo corretto e quello dell'output dei servizi.

Nel capitolo seguente mostrerò i risultati di questo metodo attraverso grafici che mostrano i punteggi di similarità per ogni servizio o attraverso esempi di traduzione.

[31] Link Dataset su Kaggle:
https://www.kaggle.com/datasets/hgultekin/paralel-translation-corpus-in-22-languages ,
Link ufficiale UE Acquis Communautaire:
https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory_en

4 Risultati

4.1 Analisi qualitativa

In questa sezione porterò alcuni esempi che mostrano le differenze di traduzione o errori che sono stati evidenziati durante la verifica manuale delle traduzioni.

Uno dei primi problemi trovati è sulla traduzione letterale di Taro:


```
"title": "Volkswagen vuole quotare Porsche in borsa",
"language": "IT",

"argos": "Volkswagen wants to quote Porsche in bag ",
"deepl": "Volkswagen wants to list Porsche on the stock exchange",
"google": "Volkswagen wants to list Porsche on the stock exchange",
"aws": "Volkswagen wants to list Porsche on the stock exchange",
"microsoft": "Volkswagen wants to list Porsche on the stock exchange",
"ibm": "Volkswagen wants to list Porsche on a stock exchange ",
"modernmt":  "Volkswagen wants to list Porsche on the stock exchange",
"openai": "Volkswagen wants to list Porsche on the stock exchange",
```


Da notare che la parola italiana "borsa" ha due significati: accessorio di abbigliamento e il mercato in cui avvengono le contrattazioni per strumenti finanziari. Questa notizia parla di quotare in borsa Porsche, usando evidentemente il secondo significato. Argos sbaglia quindi a capire il contesto della frase traducendo "borsa" in "bag".
Da notare invece che tutti gli altri servizi traducono diversamente da Argos, ma nello stesso identico modo tra loro.
La stessa identica frase in un prove successive viene tradotta diversamente da diversi servizi.


```
"title": "Volkswagen vuole quotare Porsche in borsa",
"language": "IT",
"argos": "Volkswagen wants to quote Porsche in stock",
"deepl": "Volkswagen wants to list Porsche on the stock exchange",
"google": "Volkswagen wants to list Porsche on the stock exchange",
"aws": "Volkswagen wants to list Porsche on the stock market",
"microsoft": "Volkswagen wants to list Porsche on the stock exchange",
"ibm": "Volkswagen wants to list Porsche on a stock exchange ",
"modernmt": "Volkswagen wants to list Porsche on the stock exchange",
"openai":  "Volkswagen wants to list Porsche on the stock exchange",
```


Come si può vedere la stessa frase viene in questo esempio tradotta diversamente da Argos in modo sbagliato dato che la parola "stock" da sola significa "azione", mentre in questo caso si sta riferendo al mercato azionario, cioè lo "stock exchange" o "stock market".
Anche la traduzione di Amazon è cambiata, usando "stock market" al posto di "stock exchange". Questa traduzione è comunque corretta.

```
"subtitle": " Il prezzo di oro, petrolio e gas è in forte aumento, mentre il rublo è
crollato rispetto al dollaro e ci sono sensibili perdite per le borse\t",
"language": "IT",
"argos": " The price of gold, oil and gas is rising sharply, while the ruble collapsed
 compared to the dollar and there are sensitive losses for bags",
"deepl": " Gold, oil and gas prices are rising sharply, while the ruble has collapsed
against the dollar and there are significant losses for stock markets",
"google": " The price of gold, oil and gas is rising sharply, while the ruble has
collapsed against the dollar and there are significant losses for the stock markets",
"aws": " The price of gold, oil and gas has risen sharply, while the ruble has fallen
against the dollar and there are significant losses for the stock exchanges\t",
"microsoft": "translated_subtitle": " The price of gold, oil and gas is rising sharply
, while the ruble has collapsed against the dollar and there are significant losses
for the stock markets\t",
"ibm": " The price of gold, oil and gas is rising sharply, while the ruble has
collapsed against the dollar and there are sensitive losses for the stock exchanges.\t
",
"modernmt": " The price of gold, oil and gas is rising sharply, while the ruble has
collapsed against the dollar and there are significant losses for the stock markets",
"openai": " The price of gold, oil, and gas is sharply rising, while the ruble has
collapsed against the dollar and there are significant losses for the stock markets"
```


In questo caso si tratta di un sottotitolo, infatti è più lungo e anche in questa frase, si verifica lo stesso errore sempre da parte di Argos traducendo borse con "bags" mentre tutti gli altri servizi traducono correttamente comprendendo il contesto della frase.


```
"title": "Borsa: Hong Kong negativa, apre a -0,38%",
"language": "IT",
"argos": "Bag: Hong Kong negative, opens to -0,38%",
"deepl": "Borsa: Hong Kong down -0.38%",
"google": "Stock market: Hong Kong negative, opens at -0.38%",
"aws": "Stock market: Hong Kong negative, opens at -0.38% ",
"microsoft": "Stock Exchange: Hong Kong negative, opens at -0.38%",
"ibm":  "Stock exchange : Hong Kong negative, opens at -0.38% ",
"modernmt": "Stock market: Hong Kong negative, opens at -0.38%",
"openai":  "Stock market: Hong Kong opens negatively, down 0.38%"
```


Questa è una notizia presa dalla sezione Economia di Ansa utilizzata nel testbed di economia.
Argos anche in questo caso sbaglia nuovamente la traduzione, ma la cosa interessante è che anche DeepL sbaglia la traduzione non traducendo la parola borsa.
Viceversa con la notizia sottostante molto simile e dello stesso tipo della precendete Argos e DeepL traducono correttamente, mentre IBM presenta dei problemi:


```
"title": "Borsa: Tokyo, apertura in rialzo (+0,42%)",
"language": "IT",
"argos": "Stock exchange: Tokyo, opening up (+0.42%)",
"deepl": "Stock Exchange: Tokyo, opening up (+0.42%)",
"google": "Stock market: Tokyo, opening higher (+0.42%)",
"aws": "Stock market: Tokyo, opening higher (+0.42%) ",
"microsoft": "Stock Exchange: Tokyo, opening higher (+0.42%)",
"ibm": "Tokyo, Tokyo, opening higher (+ 0.42%) ",
"modernmt": "Stock Exchange: Tokyo, opening up (+0.42%)",
```

```
"openai": "Stock Market: Tokyo, opening up (+0.42%)",
```


Questo comportamento l'ho riscontrato anche in altre notizie. La prima parola della frase viene spesso tradotta male dal servizio di IBM:


```
            "title": "Quanto è dipendente la Germania dal gas russo",
            "subtitle": " Tantissimo: ed è il motivo per cui in questi giorni si sta
opponendo a nuove sanzioni europee che riguardino il gas ",
            "ibm": {
              "translated_title": "How much is Germany dependent on Russian gas ",
              "translated_subtitle": " TMost : and it is the reason why these days are
 opposing new European sanctions that concern gas "
            }
```


La maggior parte dei servizi sono in grado di tradurre degli acronimi con il corrispettivo acronimo tradotto in inglese senza utilizzare nessun glossario.


```
"title": "Des impacts \"irréversibles\" : le Giec alerte sur les effets du changement
climatique",
"subtitle": "\nLes scientifiques du Giec publient, lundi, le deuxième  volet de leur
sixième rapport. Dans ce nouvel opus, ils abordent les effets du changement climatique
 sur les sociétés humaines et les écosystèmes et rappellent la nécessité de renforcer
les moyens de s'y adapter, en multipliant les mesures de réduction des émissions de
CO2.\n",
"language": "FR",
"argos": {
  "translated_title": "\"irreversible\" impacts: Giec alerts about the effects of
climate change",
  "translated_subtitle": "\nGiec scientists publish the second part of their sixth
report on Monday. In this new opus, they address the effects of climate change on
human societies and ecosystems and recall the need to strengthen ways to adapt to it,
by multiplying CO2 emission reduction measures.\n"
},
"deepl": {
  "translated_title": "Irreversible\" impacts: IPCC warns of the effects of climate
change",
  "translated_subtitle": "\nOn Monday, the scientists of the IPCC published the second
 part of their sixth report. In this new opus, they discuss the effects of climate
change on human societies and ecosystems, and reiterate the need to strengthen the
means of adaptation, by stepping up measures to reduce CO2 emissions.\n"
},
"google": {
  "translated_title": """Irreversible impacts: the IPCC warns of the effects of
climate change",
  "translated_subtitle": " IPCC scientists published the second part of their sixth
report on Monday. In this new opus, they address the effects of climate change on
human societies and ecosystems and recall the need to strengthen the means to adapt to
 it, by increasing measures to reduce CO2 emissions."
},
"aws": {
  "translated_title": """Irreversible impacts: the IPCC alerts on the effects of
climate change",
  "translated_subtitle": "\nOn Monday, IPCC scientists are publishing the second part
of their sixth report. In this new opus, they discuss the effects of climate change on
```

```
 human societies and ecosystems and recall the need to strengthen the means to adapt
to them, by multiplying measures to reduce CO2 emissions.\n"
},
"microsoft": {
  "translated_title": "\"Irreversible\" impacts: the IPCC warns of the effects of
climate change",
  "translated_subtitle": "\nOn Monday, IPCC scientists published the second part of
their sixth report. In this new opus, they address the effects of climate change on
human societies and ecosystems and remind us of the need to strengthen the means to
adapt to it, by multiplying measures to reduce CO2 emissions.\n"
},
"ibm": {
  "translated_title": "\"Irreversible\" impacts: the IPCC warns on the effects of
climate change ",
  "translated_subtitle": " The Giec scientists publish the second part of their sixth
report on Monday. In this new opus, they discuss the effects of climate change on
human societies and ecosystems, and remind us of the need to strengthen the means to
adapt to it, by increasing measures to reduce CO2 emissions.\n"
},
"modernmt": {
  "translated_title": "\"Irreversible\" impacts: the IPCC warns of the effects of
climate change ",
  "translated_subtitle": "\nOn Monday, IPCC scientists published the second part of
their sixth report. In this new opus, they address the effects of climate change on
human societies and ecosystems and recall the need to strengthen the means to adapt to
 it, by multiplying measures to reduce CO2 emissions."
},
"openai": {
  "translated_title": "\"Unprecedented\" impacts: IPCC warns about the effects of
climate change ",
  "translated_subtitle": "\nOn Monday, the IPCC scientists published the second part
of their sixth report. In this new opus, they address the effects of climate change on
 human societies and ecosystems and emphasize the need to enhance adaptation measures
by implementing multiple CO2 emission reduction strategies."
  }
```


In questa notizia è presente l'acronimo francese Giec: Groupe d'experts Intergouvernemental sur l'Évolution du Climat che è il nome di un foro scientifico organizzato dalle Nazioni Unite avente un acronimo diverso a seconda della lingua. Per esempio in inglese è IPCC: Intergovernmental Panel on Climate Change.
Come si può notare Argos non traduce l'acronimo nè nel titolo nè nel sottotitolo, mentre IBM traduce in IPCC per il titolo ma mantiene Giec nel sottotitolo.

4.2 Notizie generali

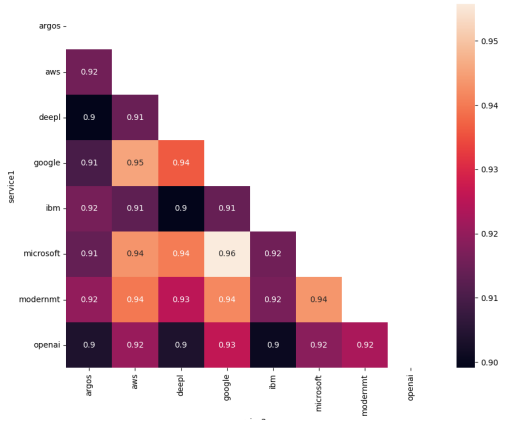

Figura 5: Punteggi medi di coseno di similarità per i titoli

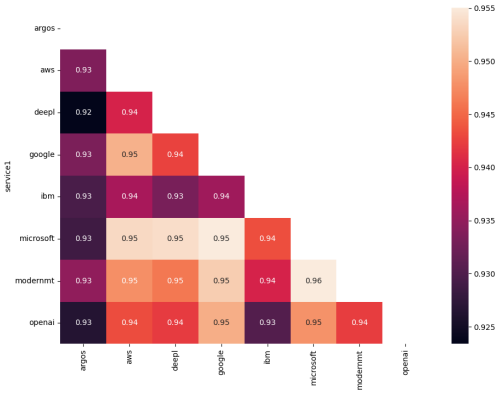

Figura 6: Punteggi medi di coseno di similarità per i sottotitoli

Le Figure 5 e 6 rappresentano un grafico di tipo heatmap che indica il punteggio medio di ogni combinazione di servizi.
La figura 5 rappresenta quelli dei titoli mentre la 6 quelli dei sottotitoli. In entrambi i casi i punteggi sono alti, tutti in un intervallo compreso tra 0,9 e 0,96.
Si può osservare un comportamento simile tra coppie di servizi, per esempio Microsoft-Google ha un punteggio di 0,96 nella tabella dei titoli e mantiene un punteggio simile di 0,95 nella tabella dei sottotitoli.
La coppia con il punteggio peggiore in figura 5 è Deepl-Argos per i titoli con 0,9, migliora il punteggio nella tabella dei sottotitoli a 0,92, ma è comunque uno tra i punteggi più bassi.
Come si può notare Argos è quello con i punteggi più bassi seguito da IBM.

Successivamente ho calcolato la media per ogni servizio.
In questo istogramma in figura 7 sono rappresentate le medie divise per servizi, per titolo e sottotitolo.
Come si era visto precedentemente i servizi che hanno i punteggi di coseno di similarità peggiori sono Argos e IBM. Si può notare che sia per Argos che per IBM i punteggi del titolo sono inferiori a quelli del sottotitolo, più che negli altri casi.
Per poter mostrare queste differenze minime, nell'ordine dei centesimi, il grafico parte da 0,8.

In figura 8 è presente un grafico di tipo boxplot.
Questa rappresentazione ci permette di vedere in figura 8 che per ogni servizio ci sono diversi outlier, nessuno con punteggi molto inferiori a 0,6 e gli outlier con i punteggi peggiori sono sempre sui titoli. Questo può essere dovuto dal fatto che i titoli possono avere una lunghezza molto breve e quindi comportare una differenza maggiore nelle traduzioni tra un servizio e un altro.
Un'altra cosa che si può notare è che l'interquartile dei sottotitoli è inferiore a quello dei titoli, questo conferma l'idea che con testi più lunghi le differenze di traduzione si riducono. Questo è dato anche da come viene calcolato il coseno di similarità.

I due grafici, figura 9 e figura 10 utilizzano la similarità di spacy.
Se confrontiamo l'istogramma in figura 9 con quello in figura 7 possiamo notare che il punteggio del titolo è inferiore a quello del sottotitolo per tutti i servizi, ma nonostante questo i punteggi sono alti, tutti superiori a 0,9. Infatti anche questo grafico utilizza come valore iniziale 0,8.
Anche in questo boxplot in figura 10 possiamo notare che gli outlier con punteggi bassi ci sono solo per i titoli mentre i sottotitoli hanno degli intervalli interquartili estremamente sottili rispetto ai titoli.
Posso presumere che con l'aumentare della lunghezza il punteggio di semantica calcolato da spacy si uniformi.

In figura 11 si possono vedere le medie per ogni servizio, calcolate sia sui titoli che sui sottotitoli e divise a seconda dell'algoritmo utilizzato.
Il punteggio più basso è quello di Hamming, come si poteva prevedere dato che spesso le traduzioni utilizzano le stesse parole, ma in ordine diverso questo provoca una distanza tra i testi alta.
Gli algoritmi in cui la posizione delle parole non ha effetto sul calcolo della similarità, cioè coseno e spacy, hanno punteggi alti ed estremamente simili per ogni servizio.
LCSS mostra un punteggio vicino a 0,8 questo significa che le frasi hanno la stessa struttura generale.

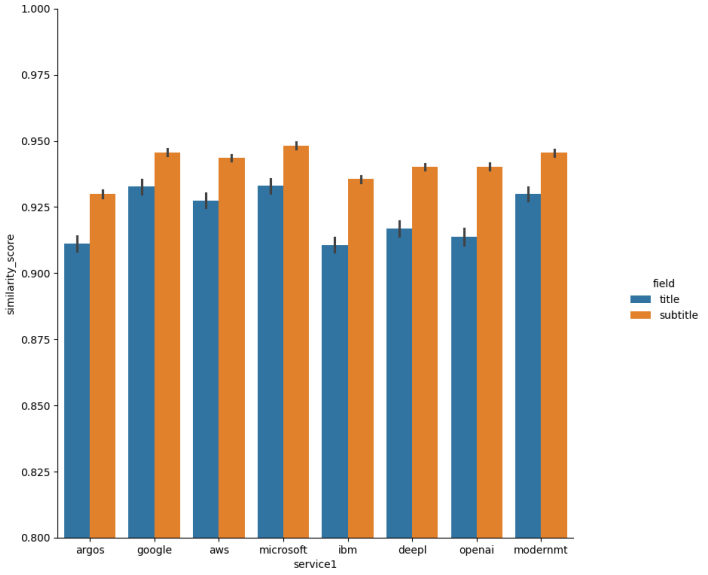

Figura 7: Medie del coseno di similarità per titoli e sottotitoli

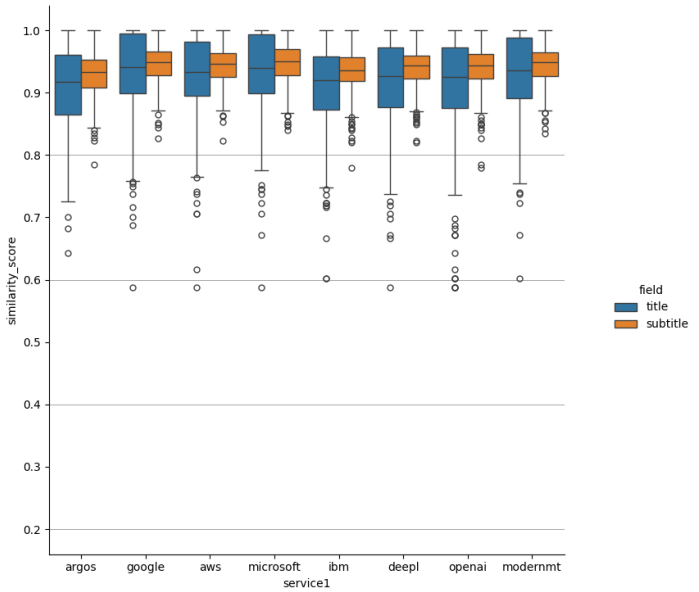

Figura 8: Boxplot delle medie del coseno di similarità per titoli e sottotitoli

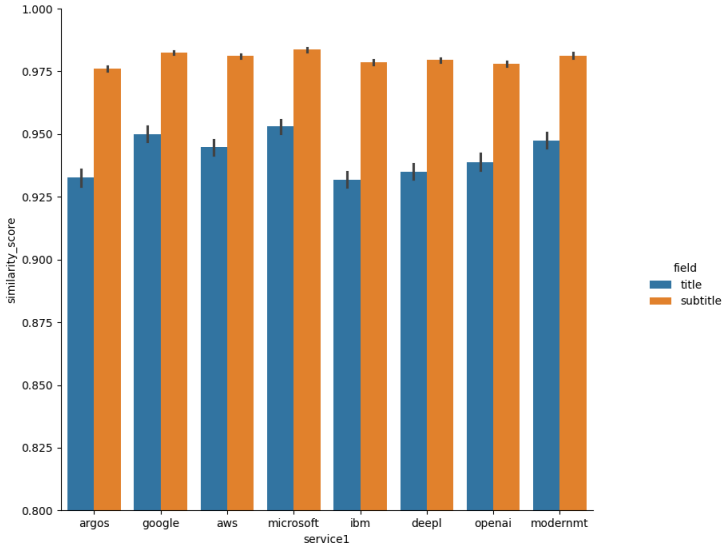

Figura 9: Medie di similarità di spacy per titoli e sottotitoli

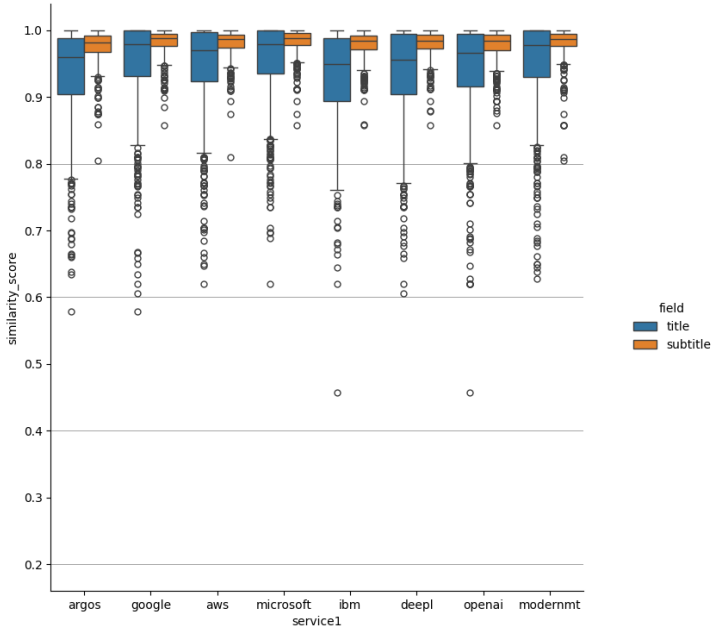

Figura 10: Boxplot delle medie della similarità di spacy per titoli e sottotitoli

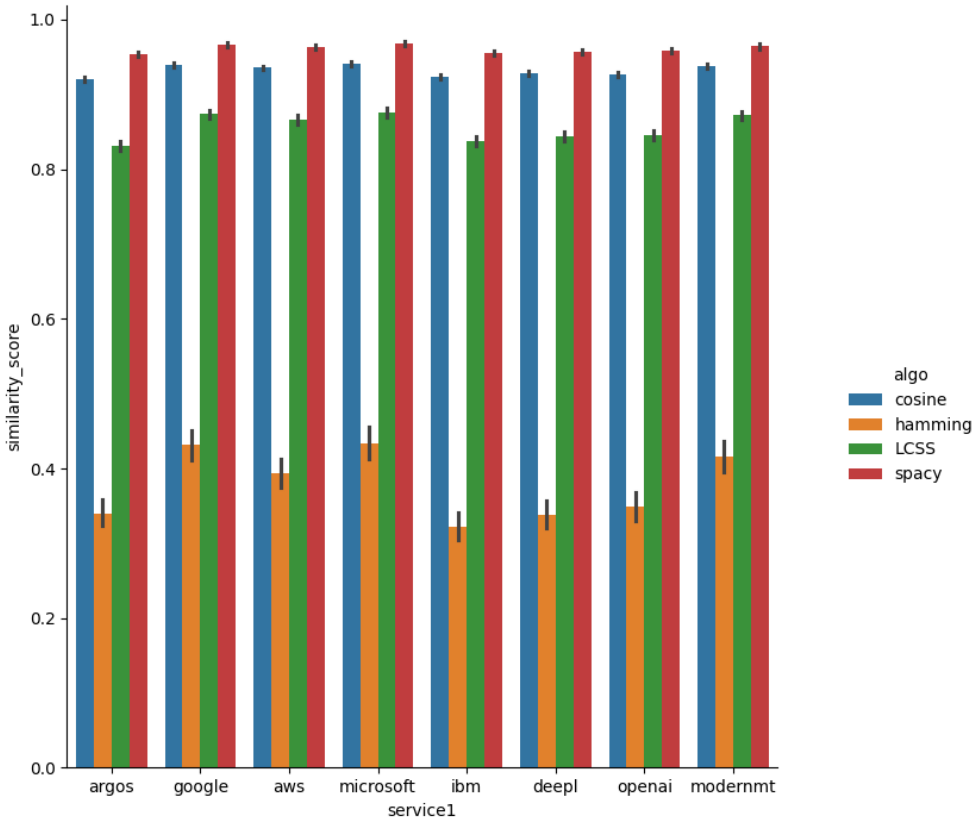

Figura 11: Confronto dei punteggi medi normalizzati con ogni algoritmo per ogni servizio, su titoli e sottotitoli

4.3 Notizie economiche

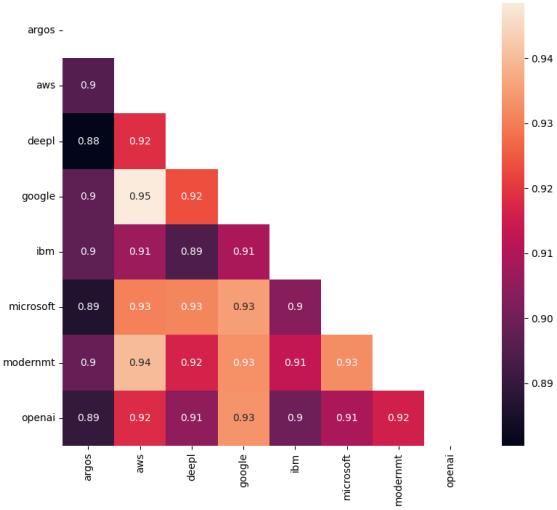

Figura 12: Punteggi medi di coseno di similarità per i titoli

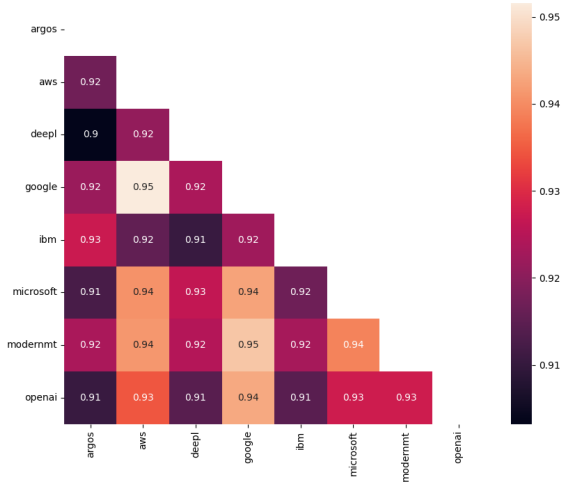

Figura 13: Punteggi medi di coseno di similarità per i sottotitoli

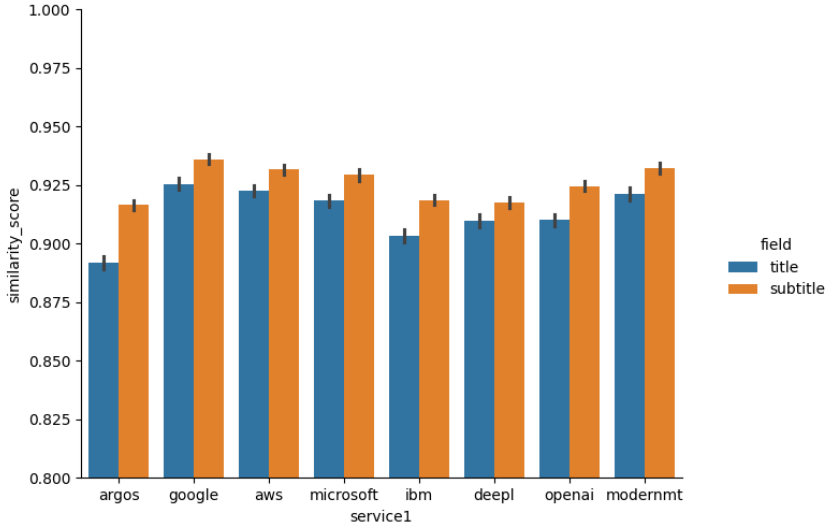

Figura 14: Punteggi medi del coseno di similarità per titoli e sottotitoli

Confrontate alla sezione precedente queste heatmap mostrano una tendenza simile, possiamo però notare punteggi per i titoli più bassi nella coppia deepl-argos, che ha il punteggio più basso in assoluto con 0,88 e in generale punteggi mediamente più bassi anche se solo di 0.01. Per i sottotitoli i punteggi sono invece comparabili con quelli delle notizie generali.

Confrontando figura 14 con figura 7 si nota come ci sia una differenza di punteggio maggiora tra titoli e sottotitoli. Infatti in questo caso i sottotitoli hanno sempre un punteggio inferiore.

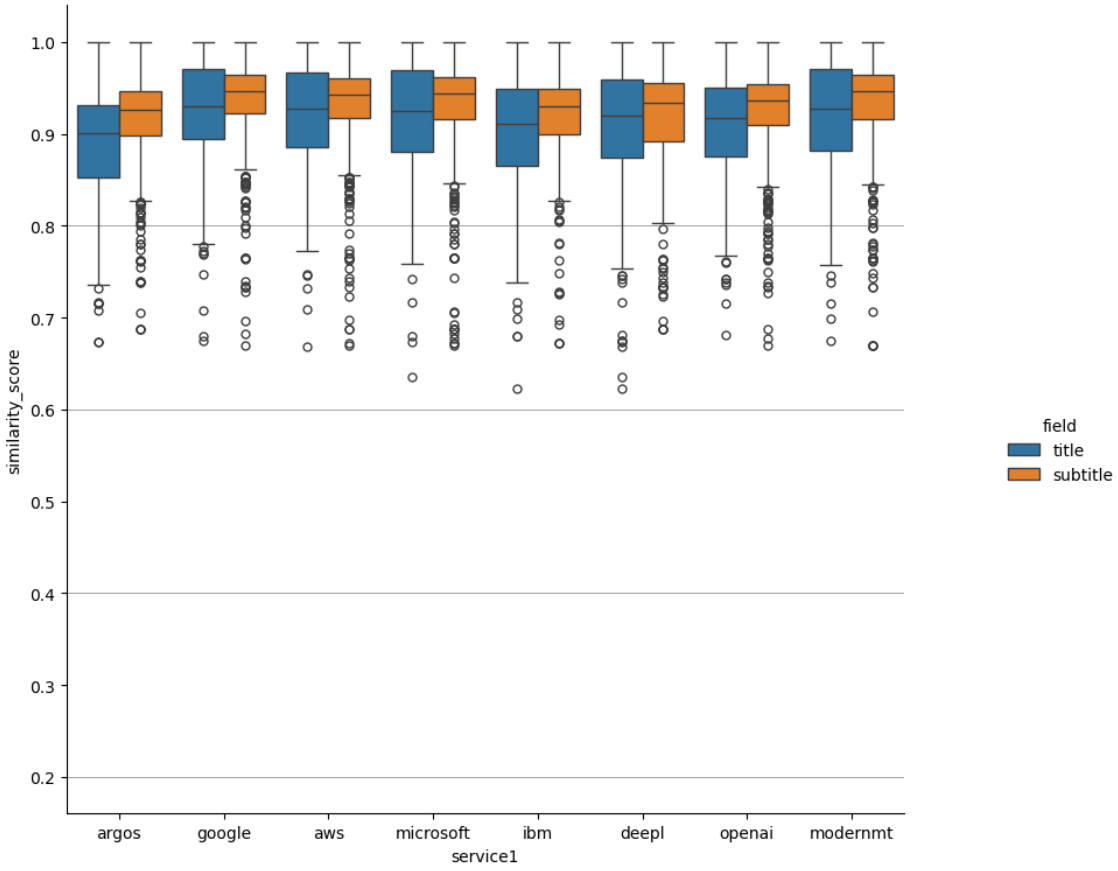

Figura 15: Boxplot del coseno di similarità per titoli e sottotitoli

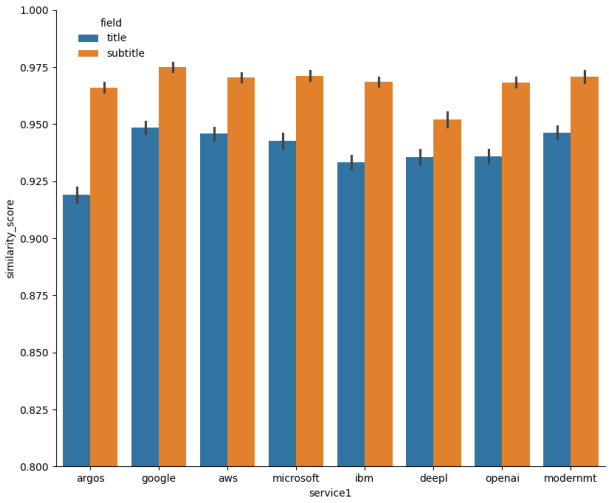

Figura 16: Medie della similarità di spacy per titoli e sottotitoli

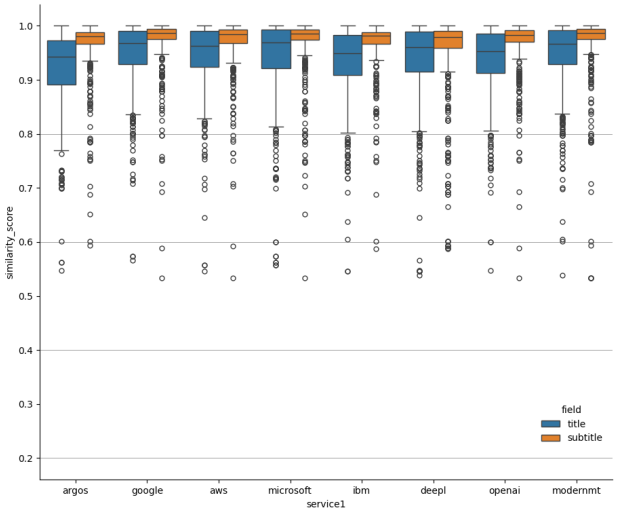

Figura 17: Boxplot della similarità di spacy per titoli e sottotitoli

Rispetto ai boxplot in figura 8 sono presenti molti più outlier. I sottotitoli ottengono punteggi più alti rispetto ai titoli, cosa che non succedeva così chiaramente per il caso generale.
Come si è visto anche negli esempi nella sezione 4.1 i titoli di questo genere di notizie sono molto corti e hanno spesso errori o traduzioni diverse tra loro e ciò viene evidenziato dal grafico.

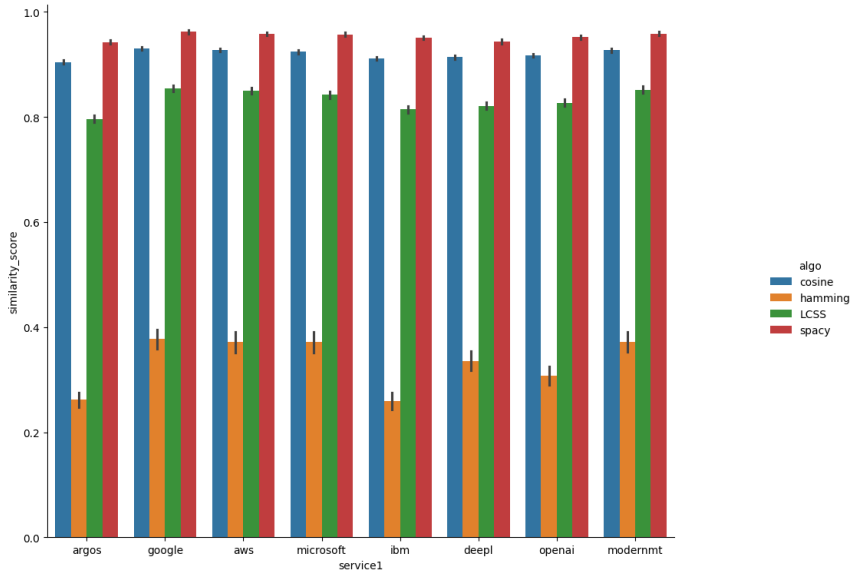

Figura 18: Medie delle similarità per ogni algoritmo

Per quanto riguarda la similarità di spaCy è paragonabile a quella del caso generale, con i punteggi dei sottotitoli superiori a quelli dei titoli.

Più interessante invece è il grafico in figura 17 dove si può notare un maggior numero di outlier rispetto al caso generale, dovuto al fatto che l'intervallo interquartile sia dei titoli che dei sottotitoli è più piccolo. La distanza tra le mediana dei sottotitoli è sempre maggiore di quella dei titoli per tutti i servizi a differenza del caso generale dove i due valori erano molto più vicini.

Infine nel confronto con gli altri algoritmi troviamo lo stesso pattern visto per il caso generale con alcune differenze. La principale è che la similarità di Hamming ha un punteggio mediamente inferiore rispetto al caso generale nonostante i valori degli altri algoritmi siano comparabili.

4.4 Notizie sportive

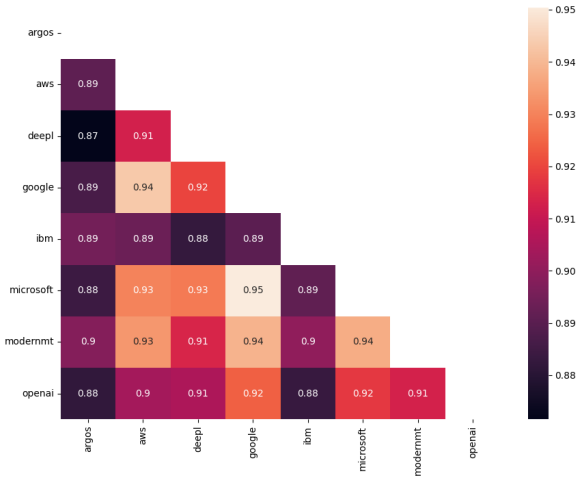

Figura 19: Punteggi medi di coseno di similarità per i titoli

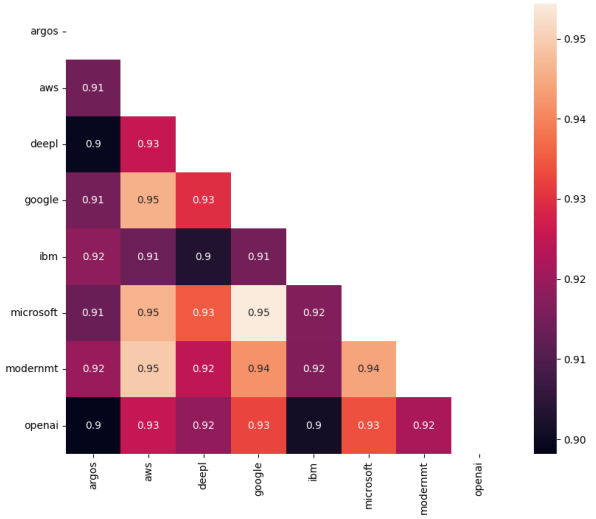

Figura 20: Punteggi medi di coseno di similarità per i sottotitoli

Confrontando 19 e 20 con le rispettive figure delle sezioni precedenti si nota una maggior somiglianza con quelle del testbed di economia avendo punteggi medi che vanno da un minimo di 0,87 a un massimo di 0,95. Nonostante ci siano delle differenze tra i due casi esse sono marginali, risulta quindi anche in questo caso che le notizie di una campo specifico ottengono punteggi medi più bassi rispetto al caso generale.

Analogamente a quello che si è visto per il testbed di economia anche per lo sport i sottotitoli hanno una mediana maggiore dei titoli e un maggior numero di outlier rispetto al caso generale.

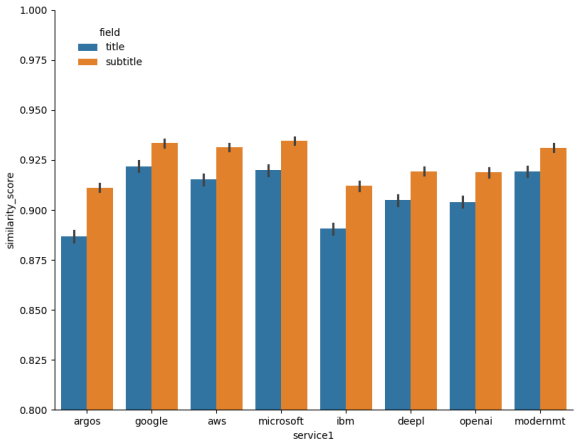

Figura 21: Medie coseno di similarità

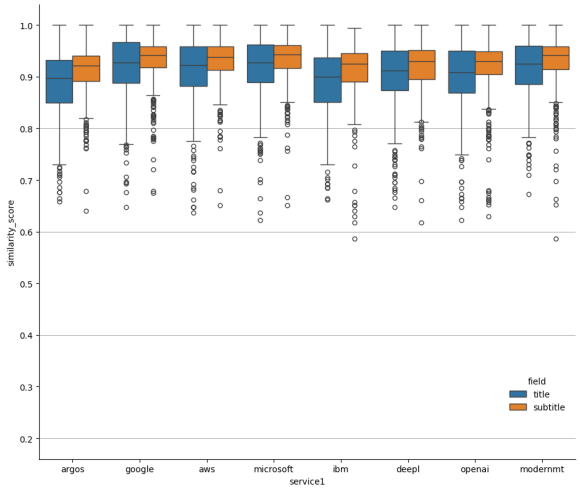

Figura 22: Boxplot coseno di similarità

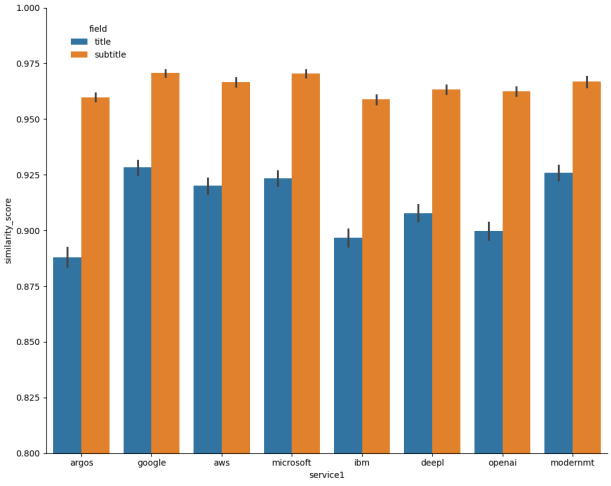

Figura 23: Media similarità di spacy

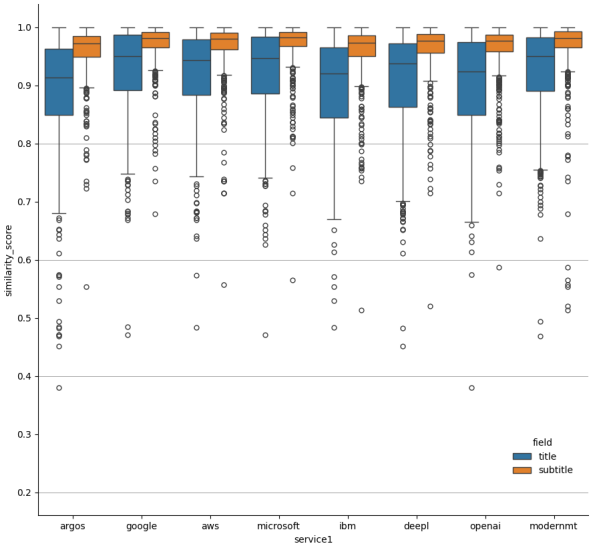

Figura 24: Boxplot similarità di spacy

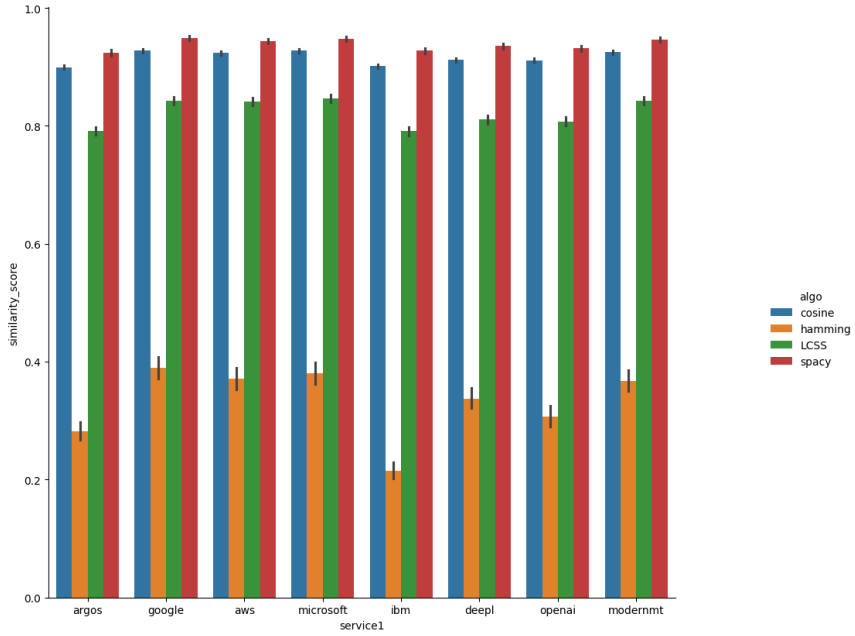

Figura 25: Medie similarità per ogni algoritmo

Per quanto riguarda la similarità di spaCy c'è una differenza maggiore tra titoli e sottotitoli sia rispetto a caso generale che a quello economico.
Si accentuano anche le differenze tra un servizio e l'altro come si può vedere in figura 23, sebbene i punteggi di tutti i servizi per i sottotitoli sono tutti compresi in un intervallo 0.96-0.97 i punteggi dei titoli variano molto a seconda del servizio.
Argos ha il punteggio peggiore, seguito da IBM e openAI.
Si può notare questo comportamento anche dal grafico di figura 24. Se si confronta quest'ultimo con quello di figura 18 si nota che gli intervalli interquartili sono più ampi sia per i sottotitoli che per i titoli, sono più ampi anche rispetto a quelli del caso generale.
Visto anche il numero di outlier, si potrebbe dire che c'è una maggiore incertezza sulla traduzione di notizie che riguardano lo sport.

Per quanto riguarda l'ultimo grafico non ci sono particolari sorprese, è del tutto simile a quello visto nella sezione precedente con punteggi Hamming più bassi del caso generale, ma con gli altri 3 algoritmi comparabili agli altri testbed.

4.5 Frasi legislative (Acquis Communautaire)

Mostro prima i risultati dei confronti tra le traduzioni dei servizi non considerando le traduzioni corrette.

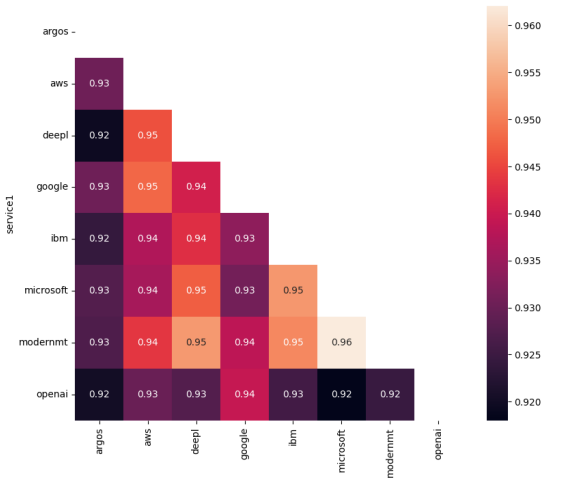

Figura 26: Punteggi medi di coseno di similarità tra servizi

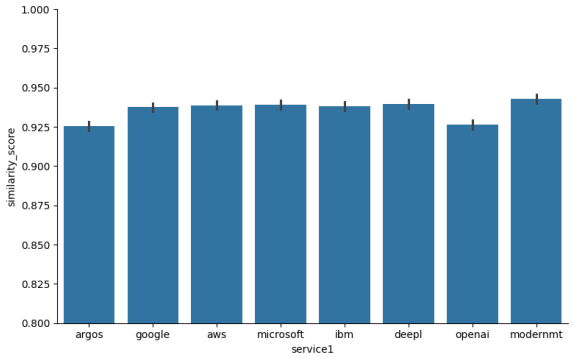

Figura 27: Medie coseno di similarità tra servizi

In questo testbed non ci sono titoli e sottotitoli quindi qua si vede una sola immagine che confronta i punteggi del coseno di similarità tra le frasi tradotte. I punteggi sono uniformi, molto simili a quelli dei sottotitoli del caso generale, figura 7.
Come si vede in figura 27 le medie dei punteggi del coseno di similarità hanno valori così vicini che sembrano tutte uguali. Tra tutte risaltano quelle di Argos e OpenAI per essere più basse anche se la differenza con le altre è minima (≈0,01).

Anche per le mediane si vede la stesso comportamento, ma differenza di quanto visto per le notizie, in questo contesto non ci sono outlier con punteggi più bassi di 0,7. Questo potrebbe essere dovuto al fatto che il linguaggio legislativo è meno propenso ad essere interpretato.

Le medie della similarità di spaCy di figura 30 presentano più differenze tra loro ed alcuni servizi hanno errori molto importanti. Si può spiegare questo comportamento guardando figura 31.
Tutti i servizi presentano un outlier con un punteggio molto basso, ma Argos, Microsoft, IBM e ModernMT hanno più outlier. Essendo il testbed composto solo da 40 frasi, questi risultati hanno un peso significativo nel calcolo della media. Le medi-

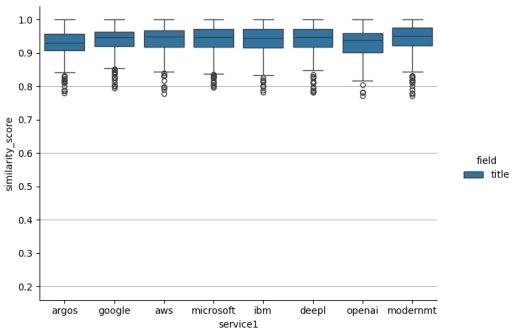

Figura 28: Boxplot coseno di similarità tra servizi

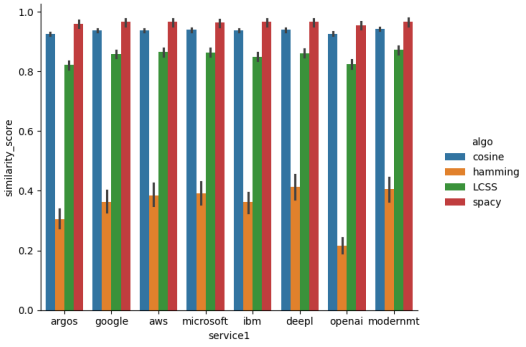

Figura 29: Medie coseno di similarità per tutti gli algoritmi tra i servizi

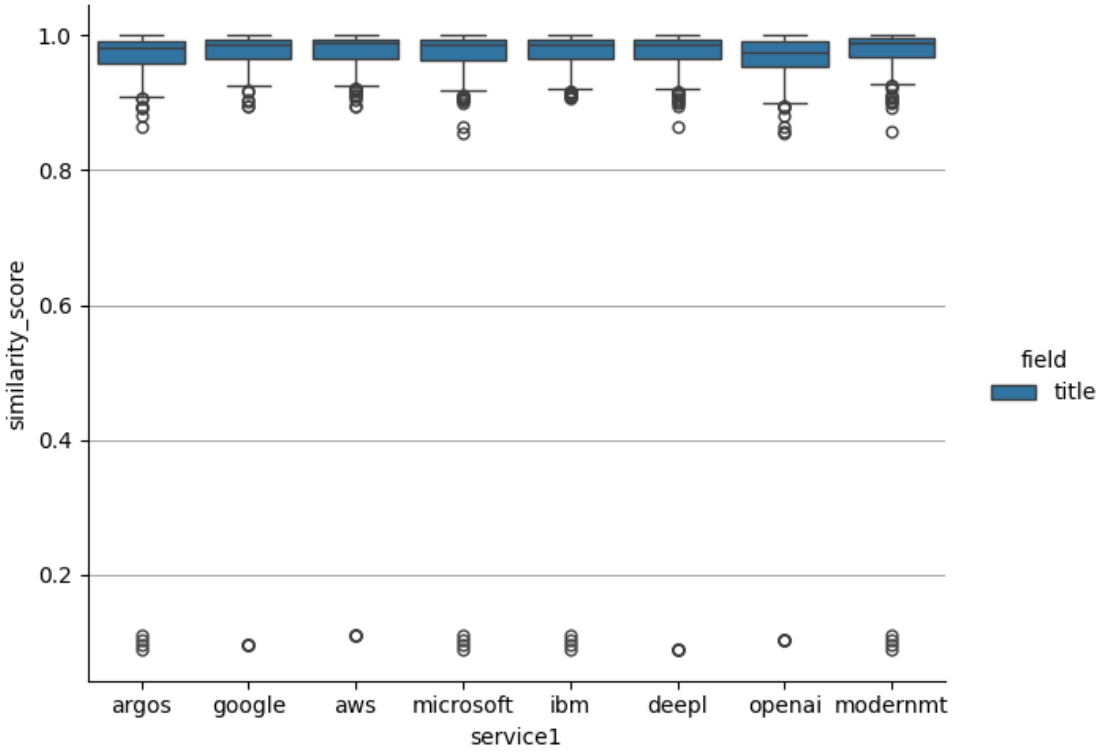

Figura 30: Boxplot della similarità di spacy tra i servizi

ane invece sono tutte allineate su punteggi superiori a 0,9 dimostrando che quasi la totalità delle frasi è stata considerata avere un significato estremamente simile tra i servizi, mentre alcuni casi hanno significati quasi completamente diversi.
Mostro di seguito come esempio uno di questi outlier.


```
"title": "La Comisión adoptará mediante actos delegados, de conformidad con el
artículo 56 y observando las condiciones establecidas en los artículos 57 y 58,
disposiciones en las que se especifique el contenido de la información que habrá de
ser objeto de intercambio en virtud del apartado 1",
"language": "ES",
"translated": "The Commission shall adopt, by means of delegated acts in accordance
with Article 56 and subject to the conditions of Articles 57 and 58, measures
specifying the content of the information to be exchanged pursuant to paragraph 1",

"argos": "The Commission shall adopt, through delegated acts, in accordance with
article 56 and noting the conditions laid down in articles 57 and 58, provisions
specifying the content of the information to be exchanged under paragraph 1",
"deepl": "The Commission shall adopt, by means of delegated acts in accordance with
Article 56 and subject to the conditions laid down in Articles 57 and 58, provisions
specifying the content of the information to be exchanged pursuant to paragraph 1.",
"google":  "The Commission shall adopt by delegated acts, in accordance with Article
56 and subject to the conditions laid down in Articles 57 and 58, provisions
specifying the content of the information to be exchanged under paragraph 1",
"aws": "The Commission shall adopt, by means of delegated acts, in accordance with
Article 56 and subject to the conditions set out in Articles 57 and 58, provisions
specifying the content of the information to be exchanged pursuant to paragraph 1",
"microsoft": "The Commission shall, by means of delegated acts, in accordance with
Article 56 and subject to the conditions laid down in Articles 57 and 58, adopt
provisions specifying the content of the information to be exchanged pursuant to
paragraph 1",
"ibm": "The Commission shall adopt, by means of delegated acts in accordance with
Article 56 and subject to the conditions laid down in Articles 57 and 58, provisions
specifying the content of the information to be exchanged pursuant to paragraph 1.",
"modernmt": "The Commission shall adopt, by means of delegated acts in accordance with
 Article 56 and subject to the conditions laid down in Articles 57 and 58, provisions
specifying the content of the information to be exchanged pursuant to paragraph 1",
"openai": "The Commission shall adopt, by means of delegated acts, in accordance with
Article 56 and in observance of the conditions laid down in Articles 57 and 58,
provisions specifying the content of the information to be exchanged pursuant to
paragraph 1."
```


In questo specifico esempio i punteggi del coseno di similarità sono compresi tra 0,7 e 0,9, con alcuni servizi che ottengo 1. Microsoft e ModernMT hanno una traduzione che utilizza le stesse parole, ma non tutte nella stessa posizione all'interno della frase.

Più interessante è il confronto fra la traduzione corretta e le traduzioni automatiche. La figura 32 mostra la similarità calcolata con spaCy tra ogni servizio e la traduzione corretta, mentre in figura 29 vi è lo stesso tipo di grafico fatto usando il coseno di similarità invece che l'analisi semantica di spaCy.
Per quanto riguarda quest'ultimo i risultati sono tutti simili tra loro, nonostante alcuni servizi abbiano punteggi più bassi di altri la differenza è minima e non è suffi-

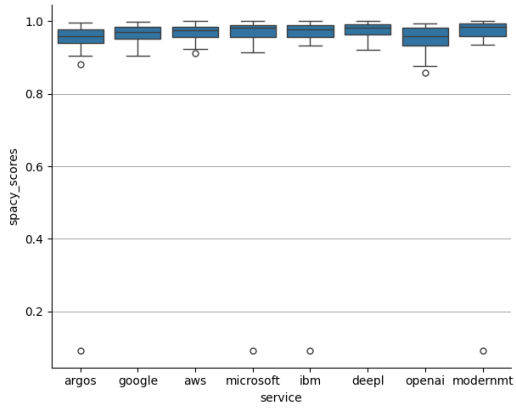

Figura 31: Medie similarità di spacy con le traduzioni corrette

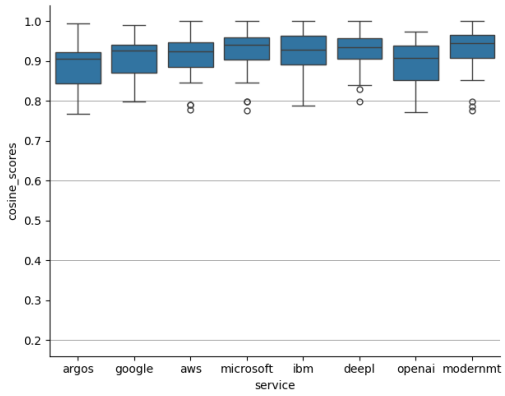

Figura 32: Boxplot coseno di similarità con le traduzioni corrette

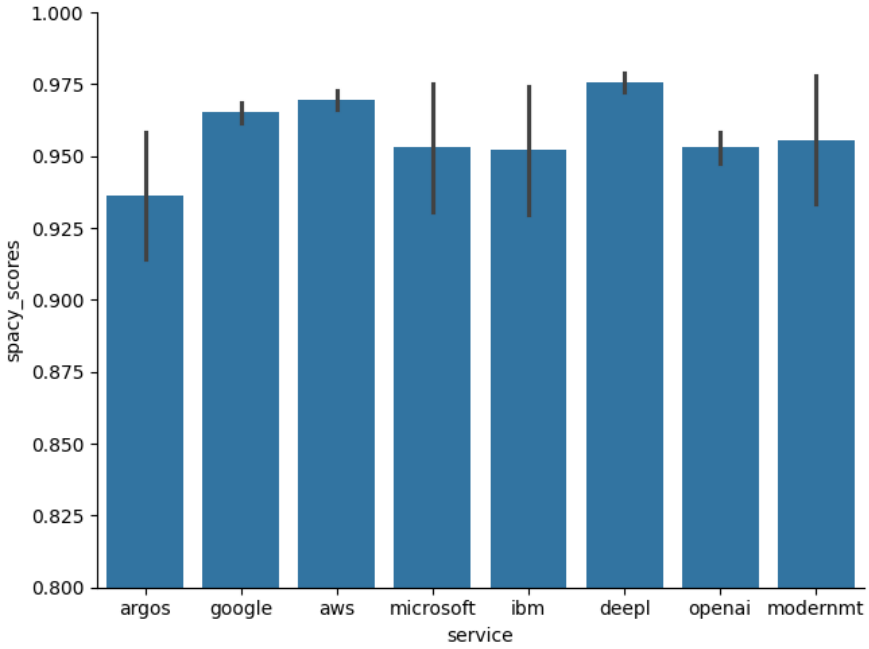

Figura 33: Similarità di spacy tra i servizi

ciente per concludere che quelle con punteggi più alti siano traduzioni migliori.

Per quanto riguarda figura 32 si può notare che sono presenti outlier per i servizi di Argos, Microsoft, IBM e ModernMT. Questo singolo outlier è sempre quello riportato precedentemente. Dato che si tratta della medesima frase ritengo che ci sia un problema su come spaCy calcoli questo punteggio in questa specifica frase e non do troppo peso a questo singolo evento, anche in questo caso le mediane sono tutte allineate, con l'unica eccezione di openAI che ha un punteggio inferiore e un'incertezza maggiore.

Nell'ultimo capitolo trarrò le conclusioni, mostrando anche i limiti di questo approccio.

5 Conclusioni

Per la traduzione di notizie i risultati mostrano che tutti i servizi sono allineati e presentano tra loro poche differenze. Risultato prevedibile poichè utilizzano tutti la stessa architettura, tranne openAI che è un LLM.
Nostante ciò alcuni servizi, come Argos e IBM, nell'analisi qualitativa hanno presentano più errori degli altri. Argos in futuro sarà probabilmente sostituito da una nuova versione e il servizio di IBM verrà completamente eliminato. Per questi motivi hanno ricevuto meno aggiornamenti rispetto agli altri, aggiornati continuamente, e questa differenza si vede nei risultati mostrati.

È interessante notare che nonostante il servizio di openAI sia basato su un LLM produce risultati del tutto paragonabili a quelli degli altri servizi.
Visto il successo degli LLM e la loro versatilità è possibile che in futuro questa tecnologia sostituisca quella più specializzata degli NMT.
Nonostante questo nel testbed sulle frasi legislative, già tradotte, OpenAI ha avuto risultati peggiori su tutti i parametri monitorati mostrando come questo servizio abbia traduzioni meno simili agli altri e alla traduzione corretta. Una possibile ipotesi è che OpenAI basandosi su un LLM generi traduzioni più libere e meno letterali rispetto agli altri servizi.
Quando si presenta un linguaggio tecnico e più preciso, come quello legislativo, OpenAI con un modello LLM crea traduzioni più libere mentre un modello NMT rimane più fedele al testo originale.

Tranne che per Argos e IBM, gli altri servizi hanno risultati talmente simili che la decisione su quale di questi utilizzare può basarsi su preferenze personali, funzionalità e costi piuttosto che sulla qualità delle traduzioni.
Infatti per poter utilizzare questi servizi in un ambiente reale o in un esperimento con un campione più ampio, vanno considerati i costi di ogni servizio.
Argos è gratuito e opensource, mentre tra gli altri Microsoft offre il prezzo minore con €9.261 per ogni milione di caratteri tradotti. Google e DeepL sono i più costosi con 20€ per ogni milione di caratteri tradotti, mentre Amazon e IBM sono nel mezzo con un costo rispettivamente di 15$ ($\approx$ 14€) e di 18€. Infine si può considerare EdenAI, in cui si possono acquistare crediti da utilizzare con qualsiasi servizio, i costi di ogni servizio sono uguali a quelli che si pagherebbero utilizzando il servizio stesso ma bisogna pagare in più un abbonamento mensile che pone comunque un limite sul numero di chiamate che si possono fare in un minuto (41$ al mese per 100 chiamate al minuto e 166$ per 300 chiamate al minuto) aumentando i costi totali e riducendo la scalabilità.

Un caso particolare è ModernMT che dal sito ufficiale offre il servizio di traduzione a 15$ per milione di caratteri mentre tramite EdenAI costa 8$.

Bisogna considerare anche i limite di questa analisi poichè il campione è ridotto, in particolare quello del Golden Standard con solo 40 frasi. Anche il numero di lingue è limitato a 4 poiché sono quelle che analizza TARO.
In possibili sviluppi futuri si potrebbe migliorare questa analisi per prima cosa ampliando il campione di partenza e aumentando il numero di lingue tradotte, in secondo luogo utilizzare input con testi più lunghi, dato che, come si è visto precedentemente nelle differenze tra titolo e sottotitolo, le traduzioni migliorano con testi più lunghi.
Un altro aspetto da esplorare meglio è quello degli LLM, monitorare il loro avanzamento includendo altri modelli di questo tipo.

Bibliografia

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Nicola Bertoldi, Davide Caroselli, and Marcello Federico. The modernmt project. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, 2018.

[3] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederick Jelinek, John Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85, 1990.

[4] Giuseppe Carrino, Angelo Di lorio, and Gioele Barabucci. Comparison of news commonality and churn in international news outlets with taro. In *Proceedings of the 34th ACM Conference on Hypertext and Social Media*, New York, NY, USA, 2023. Association for Computing Machinery.

[5] Ulrich Germann. Progress in modernmt, a new open-source machine translation platform for the translation industry. In *20th Annual Conference of the European Association for Machine Translation*, 2017.

[6] Ulrich Germann, Eduard Barbu, Luisa Bentivogli, Nicola Bertoldi, Nikolay Bogoychev, Christian Buck, Davide Caroselli, Luis Carvalho, Alessandro Cattelan, Roldano Cattoni, et al. Modern mt: A new open-source machine translation platform for the translation industry. *Baltic Journal of Modern Computing*, 4(2), 2016.

[7] Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*, 2023.

[8] Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. The OpenNMT neural machine translation toolkit: 2020 edition. In Michael Denkowski and Christian Federmann, editors, *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 102–109, Virtual, October 2020. Association for Machine Translation in the Americas.

[9] Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. OpenNMT: Neural machine translation toolkit. In Colin Cherry and Graham Neubig, editors, *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 177–184, Boston, MA, March 2018. Association for Machine Translation in the Americas.

[10] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[11] Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In Philipp Koehn, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore, December 2023. Association for Computational Linguistics.

[12] Shereen A Mohamed, Ashraf A Elsayed, YF Hassan, and Mohamed A Abdou. Neural machine translation: past, present, and future. *Neural Computing and Applications*, 33:15919–15931, 2021.

[13] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[14] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.