UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

RELAZIONE

Addestramento di un NER in *Voci della Grande Guerra*

Candidato: *Clara D'Apoli*

Relatore: *Alessandro Lenci*

Correlatore: *Felice Dell'Orletta*

Anno Accademico 2017-2018

Indice generale

1 Introduzione

“La Prima Guerra Mondiale rappresenta un punto di riferimento cruciale nella storia dell'umanità. Ha cambiato il destino di intere generazioni e le sue conseguenze geopolitiche influenzano ancora il mondo contemporaneo.” (Lenci et al., 2018, p. 101). La celebrazione del Centenario della Grande Guerra ha rappresentato un’importante occasione per la realizzazione del progetto *Voci della Grande Guerra* (VGG), che grazie alla fruttuosa collaborazione di storici e linguisti, vuole essere esempio virtuoso delle potenzialità della linguistica computazionale per le discipline umanistiche digitali. La creazione di un corpus di testi digitali selezionati dagli esperti per essere rappresentativo dei diversi modi di esperire e descrivere la guerra da parte dei suoi protagonisti, è senz’altro una possibilità senza precedenti di ottenere una visione multidimensionale e multiprospettica di eventi bellici. Cogliere questa opportunità richiede metodi avanzati per l'analisi semantica automatica delle fonti digitali, che ritroviamo negli strumenti del Natural Language Processing. In particolare, per far fronte a uno degli obiettivi finali del progetto, quello di Information Extraction (IE), e per ovviare sempre più alle problematiche di adattamento di dominio linguistico e testuale dei testi del primo Novecento, con il lavoro sviluppato in questo documento verrà sperimentato l’addestramento di un algoritmo supervisionato di Machine Learning (ML) per la classificazione di entità nominate nel corpus VGG, che nel corso dell’esperimento chiameremo “VGG NER”. Il confronto con altri classificatori (NER) preesistenti, tutti addestrati in maniera differente e testati sul corpus VGG, ci permetterà poi di valutarne al meglio le prestazioni grazie a un’attenta analisi dei risultati estratti, sulla base di metriche di valutazione specifiche (*Precision* e *Recall*). Infine, un accenno all’innovativa tecnica di addestramento basata sulle reti neurali solleciterà una progressiva ottimizzazione delle tecniche utilizzate e la scoperta di nuove frontiere della Linguistica Computazionale.

2 Voci della Grande Guerra

2.1 Il progetto

In occasione del Centenario della Prima Guerra Mondiale l'Università di Pisa in collaborazione con l'Istituto di Linguistica Computazionale, l'Università di Siena e l'Accademia della Crusca ha realizzato un progetto dal nome *Voci della Grande Guerra (VGG)*. Si tratta di un'iniziativa scientifica e culturale, nata dalla collaborazione di storici e linguisti, con l'obiettivo di preservare e diffondere la memoria della Prima Guerra Mondiale attraverso la creazione e pubblicazione del primo corpus digitale di testi, rappresentativo della polifonia linguistica dell'Italia al tempo della Grande Guerra. Il progetto mira, attraverso tecniche avanzate di Linguistica Computazionale, all'esplorazione approfondita di testi di diversi generi e registri testuali per classificare un'ampia gamma di fenomeni rilevanti per lo studio delle caratteristiche linguistiche dell'italiano del primo Novecento.

2.1.1 Sfide e obiettivi

I tipi di testi utilizzati da *Voci della Grande Guerra* sollevano un gran numero di sfide all'elaborazione del linguaggio naturale e ai metodi di analisi del testo:

- dati molto rumorosi;
- espressioni linguistiche dell'italiano popolare (sub-standard) o mal formate a causa della mancata alfabetizzazione del tempo;
- variazione diacronica e diafasica del linguaggio;

VGG affronta queste sfide applicando e sviluppando metodi per l'annotazione del testo e l'estrazione di informazioni dai testi digitalizzati, per arrivare a espandere ulteriormente e arricchire l'archivio testuale. Lo scopo principale del progetto riguarda proprio la questione morale di preservare la memoria storica dell'avvenimento, rendendola accessibile a un pubblico più ampio e vario, quindi creando un corpus digitale di testi italiani sulla Prima Guerra Mondiale, rappresentativo dei diversi modi di vivere e descrivere la guerra da parte dei suoi protagonisti. La semplice digitalizzazione delle fonti storiche non è sufficiente per accedere pienamente al loro contenuto, specie quello semantico, ma è possibile farlo grazie a tecniche di Natural Language Processing (NLP). Infatti, un altro obiettivo fondamentale del progetto riguarda l'annotazione dei testi digitalizzati con strumenti avanzati di NLP e

successiva revisione manuale di un sottocampione significativo e rappresentativo del Corpus. Inoltre, la metodologia automatica di analisi e annotazione renderà il Coprus VGG una piattaforma digitale aperta e continuamente espandibile per l'analisi e l'elaborazione dei testi storici.

Infine, lo scopo ultimo del progetto è quello di sviluppare un'interfaccia di navigazione online che possa sia assistere i ricercatori durante il processo di costruzione del Corpus, fornendo varie funzionalità di ricerca in supporto alla fase di correzione manuale, sia rappresentare uno strumento innovativo di esplorazione del Corpus con forme tecnologiche avanzate di visualizzazione e interrogazione delle informazioni.

2.1.2 Il Corpus e la lingua

"La Grande Guerra è la prima guerra di morte di massa, ma è anche la prima guerra di produzione di testi di massa." (Lenci et. al, 2016). In questo contesto, il Corpus VGG comprende testi tra i più vari generi e registri linguistici per un totale di circa 1 milione di tokens, distribuiti in più di 70 documenti, che rappresentano il corpus più vasto creato fino ad oggi per lo studio delle caratteristiche linguistiche dell'italiano del primo Novecento. Questo è stato bilanciato sulla base di diversi parametri e include testi scritti dal 1913 ai primi anni '20, così da coprire non solo gli anni della guerra, ma anche il contesto socio-culturale, antecedente e successivo, che l'ha contraddistinta. Come accennato, la considerevole varietà testuale è rappresentante della polifonia delle persone che sono state interessate dalla guerra: la voce tecnica dei giornali e quella informale delle lettere, la voce composta delle élite degli intellettuali e quella popolare, la voce del consenso e quella del dissenso, la voce ufficiale della propaganda e quella dei soldati, questi ultimi spesso cimentatisi per la prima volta nell'esperienza della scrittura, per dare un senso agli eventi drammatici e dirompenti a cui hanno preso parte. Sebbene alcuni archivi digitali di testi italiani risalenti alla Grande Guerra siano già disponibili da tempo, essi sono solitamente limitati a un solo genere testuale, principalmente diari; è, invece, grazie a questa vasta gamma di registri, generi testuali e varietà linguistiche, che è stato possibile massimizzare la rappresentatività del Corpus VGG rispetto alle varie prospettive sulla guerra. Inoltre, i testi digitalizzati sono stati abbinati alle immagini scansionate dei documenti originali, per garantire un parallelo controllo filologico delle fonti.

La realizzazione del Corpus non è stata facile, anche perché l'Italia è notoriamente in ritardo rispetto agli altri paesi interessati dalla Guerra nel processo di digitalizzazione,

ma si tratta di un'impresa estremamente rilevante sia dal punto di vista storico che linguistico. Se la Prima Guerra Mondiale è un evento abbastanza noto, molto meno lo sono le diverse prospettive narrative ed esperienziali di questa: i testi prodotti in quel periodo sono stati un'arma fondamentale per plasmare le immagini della guerra e persuadere le persone ad accettarla o rifiutarla. È proprio a tal proposito che il Corpus VGG offre agli storici un nuovo strumento digitale attraverso il quale esplorare questa ricchezza di voci estremamente diverse, e spesso dimenticate.

I linguisti hanno sempre attribuito una funzione molto importante alla Grande Guerra come momento decisivo nel processo che porta all'unificazione linguistica dell'Italia (De Mauro 1963), perché persone provenienti da ogni parte della penisola furono costrette a vivere insieme e, quindi, a usare la lingua nazionale per comunicare tra loro e con ufficiali, che avevano una padronanza della lingua sicuramente maggiore. Il confronto tra le varietà linguistiche nel Corpus VGG si pone, quindi, come obiettivi quelli di condurre a una più profonda comprensione di questi problemi di comunicazione, fornire nuove prove su come le difficoltà linguistiche siano state superate in circostanze così tragiche e consentire agli studiosi di studiare l'influenza dei modelli retorici e letterari sull'italiano standard.

2.2 Il flusso di lavoro

Il progetto *Voci della Grande Guerra* è stato realizzato con un flusso di lavoro articolato in diverse fasi per una durata complessiva di 24 mesi.

2.2.1 Acquisizione dei testi

La prima fase del lavoro consiste nel processo di acquisizione dei testi che, come precedentemente accennato, viene avviata con la selezione delle situazioni comunicative rilevanti che caratterizzano il linguaggio del tempo. I registri linguistici comprendono:

- la lingua ufficiale militare (bollettini di guerra, libri di strategia militare, documenti di propaganda e registri marziali di corte);
- la lingua della classe media (diari ufficiali, memorie, ecc.);
- la lingua popolare (lettere, diari, memorie);
- la lingua della classe politica (procedimenti parlamentari, discorsi ufficiali);
- la lingua dell'élite intellettuale (opuscoli, riviste, ecc.);

- la lingua standard dell'opinione pubblica (articoli di giornale, riviste, notizie, ecc.).

Molte delle fonti sono state facilmente recuperate da altri progetti avviati precedentemente (ad es. *Memorie di Guerra*), pubblicazioni e accordi con entità storico-culturali, mentre i problemi relativi al copyright sono stati attentamente valutati e affrontati.

Segue una fase di scansione dei testi non ancora disponibili in formato digitale, a cui ha contribuito anche la community di WikiSource, grazie all'utilizzo di scanner ad alta risoluzione, per poi procedere con il momento dell' analisi mediante il software di riconoscimento ottico dei caratteri (OCR). Durante questa fase, per aumentare la precisione del riconoscimento è stato adottato un sistema che combina le tecniche più avanzate di allineamento dell'output di più OCR.

Infine, l'output definitivo viene controllato e corretto manualmente per poi essere opportunamente codificato nel formato standard TEI-XML.

2.2.2 Analisi linguistica

La fase di analisi linguistica comprende l'annotazione automatica, la revisione manuale e l'estrazione di infomazioni. In primis i testi digitalizzati vengono esportati e sottoposti a un'elaborazione computazionale di Natural Language Processing (NLP), che viene avviata con l'annotazione automatica dei dati linguistici, quindi lemmatizzazione e analisi morfo-sintattica, e metalinguistici. La funzione di tale Trattamento Automatico della Lingua (TAL) è quella di annotare i testi con metadati semantici che arricchiscono il loro valore informativo, moltiplicando le possibilità e le modalità di accesso ai loro contenuti avanzati. Segue una fase di estrazione delle informazioni semantiche mediante un modulo di Named Entity Recognition, in cui vengono riconosciute e annotate entità nominate di persone, luoghi e organizzazioni. Successivamente un sottoinsieme del Corpus, un campione significativo di circa 650.000 tokens, è stato corretto manualmente e arricchito con ulteriori metadati mediante uno strumento di correzione appositamente progettato per questo scopo, andando così a costituire il cosiddetto Gold Standard, che verrà riutilizzato in fase di re-training (addestramento).

2.2.3 Pubblicazione web

La fase finale del progetto ha visto la messa a disposizione di alcuni strumenti online per l'esplorazione del Corpus; in particolare, è stata sviluppata una piattaforma

software che fornisce funzionalità utili alla ricerca interna a esso. Il tool comprende un modulo di back-end per supportare la correzione di testi digitalizzati e annotati automaticamente, e un modulo front-end per la visualizzazione e interrogazione delle informazioni. Quest'ultimo è interattivo e ben strutturato:

- è possibile impostare dei criteri di ricerca (per documento, per anno, per genere testuale, per autore e professione di questo), che vengono utilizzati per la ricerca avanzata di n-grammi e di termini nei testi;
- la ricerca dei termini prevede un'ulteriore selezione di parametri, per lemma o per forma, grazie ai quali vengono visualizzate tutte le occorrenze dell'espressione nei testi, di cui è possibile consultare i relativi dettagli bibliografici;
- la ricerca per n-grammi permette di visualizzare la frequenza di una o più parole su un istogramma e di filtrare i risultati ottenuti per data o per documento;
- può essere effettuata la ricerca di entità nominate all'interno di uno specifico documento, indicando il tipo di NER, quindi Luogo, Organizzazione e Persona, e digitandone il testo; verranno visualizzate la forma, il lemma e il numero di occorrenze nel documento. Inoltre, ogni tipo di NER è contraddistinto da un colore diverso, così da facilitarne il riconoscimento a un primo impatto visivo.

3 Named Entity Recognition: l'annotazione

Come precedentemente accennato, la fase di analisi linguistica automatica prevede, in seguito al processo di lemmatizzazione e di PoS tagging, il riconoscimento e l'annotazione delle Named Entities (NEs) classificate nelle categorie semantiche di Persone, Luoghi e Organizzazioni. Rappresenta, questa, una fase fondamentale per il Trattamento Automatico del Linguaggio (TAL), in quanto permette l'estrazione di informazioni semanticamente rilevanti dai testi (IE), scopo finale del progetto. In particolare, per il riconoscimento automatico di NE sono stati utilizzati algoritmi supervisionati di Machine Learning, i quali sfruttano corpora annotati a mano per la creazione di modelli di addestramento (training set) e richiedono tempi di sviluppo più brevi rispetto ai sistemi basati su regole (rule-based); non solo, questi ultimi risultano svantaggiosi anche nei compiti di Domain adaptation, cioè di adattabilità a diversi domini, generi testuali e registri linguistici, in fase di training (addestramento).

Dunque, si può facilmente intuire come l'analisi linguistica basata sull'apprendimento automatico in NLP segua il percorso di un circolo virtuoso in cui un modello di corpus annotato funge sia da punto di partenza che d'arrivo, ovvero il *gold standard* viene utilizzato per avviare la fase di training (training corpus), mentre il modello prodotto in output viene impiegato come input per il test su un nuovo campione di dati annotati (test corpus); ne segue, infine la valutazione delle performance dei sistemi di NLP adottati per entrambe le fasi.

3.1 Tagset

Il tagset scelto per l'annotazione di nomi propri nel corpus VGG è costituito da tre classi di entità: Luoghi (LOC, ad es. "Gran Sasso"), Persone (PER, ad es. "Benito Mussolini") e Organizzazioni (ORG, ad esempio, "Brigata Sassari"). In particolare, per le espressioni poliremiatiche (ad es. nomi complessi) è stato adottato lo schema di annotazione IOB (Inside Outside Begenning), secondo il quale il prefisso "B-" marca l'inizio del nome (Beginning), il prefisso "I-" la parte interna (Inside) e "O" i token che non sono entità (Outside) (Hobbs & Riloff, 2010). Ad esempio:

TOKEN	NER
Col	B-LOC
San	I-LOC
Giovanni	I-LOC

3.2 Annotazione automatica

Per l'annotazione automatica di entità nominate nel Corpus VGG è stato necessario affrontare il problema dell'arcaicità della lingua, che avrebbe notevolmente ridotto la precisione degli strumenti di annotazione basati sull'italiano standard contemporaneo. Così, Passaro e Lenci (2015) hanno adottato un identificatore di entità nominate (NER) già creato precedentemente per l'annotazione dei bollettini della Prima guerra mondiale, quindi già adattato all'italiano dei testi storici del primo Novecento. Si tratta, dunque, di un sistema di Machine Learning che consente di ovviare ai problemi di robustezza nel trattare l'input mal formato o non conforme alle regole generali della lingua italiana contemporanea, migliorando l'accuratezza dei risultati prodotti, l'efficienza nella capacità di gestire ingenti quantità di dati e l'adattabilità a diversi domini, generi e registri. Così, per velocizzare la creazione del *gold standard*, i testi sono stati prima etichettati in modo semi-automatico dal NER già esistente e quindi controllati manualmente.

3.2.1 Problematiche

Il sistema di annotazione automatica sviluppato è certamente uno strumento avanzato con tempi di sviluppo molto rapidi ma, nonostante l'adattabilità al dominio dei testi storici, come tutti gli algoritmi di apprendimento supervisionato non può fare a meno di commettere errori consistenti: il tagger, software che identifica e classifica le NE, ha una precisione stimata attorno al 98%, mentre il parser, software che analizza la struttura morfo-sintattica delle parole, lo è per circa l'85%. Per quel che concerne il tagger di NE, non è facile il riconoscimento delle entità nominate a causa di alcune ambiguità e i principali problemi riscontrati, a volte anche molto contrastanti tra loro, sono i seguenti:

- forma ortografica: parole capitalizzate a inizio frase sono state scambiate per nomi propri;
- errori sistematici: nomi di luogo e di organizzazione sono stati scambiati per nomi di persona, e viceversa, sono stati annotati erroneamente con costanza per tutta la lunghezza del testo. Ad esempio, in Monelli, i nomi di persona "Heliodora" e "Gallina" ricorrevano nel testo per ben 3 volte ciascuno, sempre con il tag B-LOC, invece che B-PER; ancora, il nome delle catene montuose "Dolomiti" compariva nel testo con una frequenza di 4 occorrenze, senza mai essere riconosciuto come entità nominata;

- errori occasionali: alcuni termini riconosciuti sempre correttamente dal NER sono stati eccezionalmente taggati in maniera errata. Ad esempio, in Monelli, il nome del politico "Zanella" su 5 occorrenze, è stato correttamente riconosciuto e annotato quattro volte come nome di persona (B-PER), ma solo una, erroneamente, come luogo (B-LOC);
- segmentazione di termini complessi: per alcuni gruppi di parole che fanno parte di un'unica entità non viene correttamente individuato il confine di questa. Ad esempio, sempre in Monelli, il luogo "Malga la Costa" compare per ben due volte, entrambe annotate erroneamente:

Malga	Malga	B-LOC
la	il	O
Costa	costa	B-PER
Malga	Malga	B-LOC
la	il	O
Costa	costa	O

Sono tutti, questi e altri ancora, errori che ci si pone di risolvere o comunque, ridurre sempre più, con la correzione manuale e il successivo riaddestramento del NER.

3.3 Correzione manuale: il *gold standard*

Per la creazione del corpus di addestramento, è necessario far fronte alle problematiche sopra citate (v.3), ovvero è necessaria la correzione manuale di quanto annotato automaticamente dal NER, un compito estremamente lento e costoso.
Per quanto riguarda le NE, la revisione ha interessato un corpus rappresentativo di 6 documenti più o meno eterogenei nel genere e nel registro:

- Discorsi di Salvemini – 2358 tokens corretti;
- Memorie di Monelli – 2413 tokens corretti;
- Diario di Martini – 35737 tokens corretti;
- Relazione della Camera Comitati segreti – 7573 tokens corretti;
- Diario di Sonnino (Vol. I e II) – 14131 tokens corretti;

I testi sono stati analizzati e corretti mediante un tool opportunamente creato, che si basa sul formato di dati "CoNLL", Conference on Natural Language Learning; in particolare, è stata utilizzata una versione riveduta del formato CoNLL-X, chiamata CoNLL-U (da Universal Dependencies), in cui le annotazioni sono codificate in file

di testo semplice (UTF-8). I documenti, esportati successivamente in formato `.conllu`, sono suddivisi nel tool in frasi composte da uno o più token, ognuno dei quali inizia su una nuova riga ed è annotato in 13 campi separati:

- ID: indice di parola (intero che parte da 1 per ogni nuova frase);
- FORM: forma del token in analisi;
- LEMMA: lemma;
- UPOS: universal part-of-speech tag;
- XPOS: part-of-speech tag specifico della lingua;
- FEATS: caratteristiche morfologiche;
- HEAD: testa del token corrente, che rappresenta un valore di ID o zero (0);
- DEPREL: relazione di dipendenza universale con HEAD;
- NER: entità nominate;
- VARIANTE LEMMA: eventuali varianti del lemma;
- ETICHETTE VARIAZIONI: la tipologia di variazione del lemma;
- NOTE: eventuali note aggiuntive.

È chiaro che diversi annotatori nel corso del progetto hanno preso a carico la correzione dei tag errati o mancanti di uno o più tra i campi sopra elencati, ma per l'obiettivo finale dell'analisi in questione, ovvero l'addestramento di un classificatore di entità nominate, ci soffermiamo sull'annotazione di queste ultime nella relativa colonna "NER". È questa una fase fondamentale nella progettazione del classificatore, poiché determina la creazione di un *gold standard* (corpus annotato a mano), necessario sia per addestrare il modello computazionale, sia per la valutazione dei sistemi di NLP, quindi nella fase di test.

Alla fine del lungo processo di revisione dei nomi propri all'interno dei testi selezionati, il totale di 2785 nomi di persona, 3141 nomi di luogo e 1145 nomi di organizzazione costituisce il cosiddetto *gold standard*, il modello di riferimento che utilizzerà l'algoritmo di ML per l'apprendimento, e che verrà quindi opportunamente distribuito tra train set e test set. È opportuno sottolineare che l'annotazione per le tre classi PER, LOC e ORG non ha riguardato soltanto entità; ad esempio per i luoghi sono stati prese in considerazione sia entità geopolitiche (ad esempio "Italia") che luoghi reali (ad esempio "Monte serra") e per persone e organizzazioni nomi istituzionali come "ministro dell'Impresa" o "ministero della Giustizia". La figura 1 mostra la distribuzione delle differenti classi di NEs nel *gold standard*.

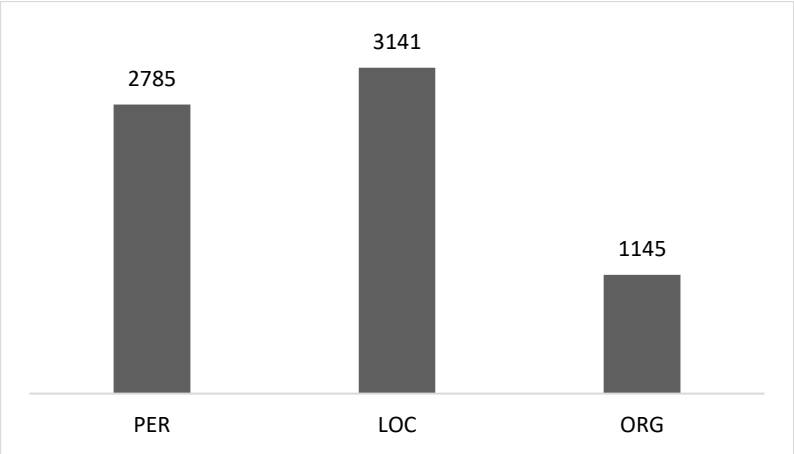

Figura 1. Distribuzione delle NEs nel *gold standard*

4 Progettazione del classificatore NER

Dopo aver preparato il corpus annotato, prima automaticamente e poi manualmente, esso è pronto ad essere impiegato per l'addestramento di un nuovo sistema di classificazione dei nomi propri utilizzando un algoritmo di apprendimento supervisionato, adattato all'italiano dei tempi della Grande Guerra. Come già accennato nelle sezioni precedenti, è stato scelto l'addestramento basato sugli algoritmi di ML in quanto risultano particolarmente efficienti ed accurati nella risoluzione di compiti di classificazione.

La creazione del NER si compone di due fasi principali:

1. Addestramento (training);
2. Analisi del modello creato (test);

L'algoritmo viene addestrato a riconoscere le categorie di entità nominate (PER, LOC, ORG) fornendogli una serie di esempi che il sistema elabora alla ricerca di una regola generale di classificazione (fase di training); una volta costruito il modello, esso viene utilizzato per identificare e classificare le nuove istanze (fase di test).

Per poter addestrare l'algoritmo di riconoscimento di entità, come già precedentemente accennato, è stato necessario creare un training set e un test set, ovvero suddividere il nostro corpus annotato (v. Figura 1) tra quello che verrà utilizzato per addestrare il classificatore, e quello che invece verrà preso in input nella fase di test per verificare l'efficacia del modello costruito in training.

Per questo lavoro, la partizione è stata estremamente rigorosa, dunque si è tenuto conto sia di uno degli obiettivi finali del progetto di *Domain adaptation* e si è cercato di mantenere una certa eterogeneità di genere, registro e tema nel training e nel test, sia delle proporzioni necessarie al corretto bilanciamento di training e test set in termini di entità. Infatti, proprio per quest'ultimo compito, è stata stimata una suddivisione della distribuzione di Named Entities variabile tra 60-70% nel corpus di training e tra 30-40% in quello di test, affinchè possano entrambi essere essere abbastanza rappresentativi del corpus VGG.

- Training set:
    - Parte di "Discorsi di Salvemini" (76%): 22 PER - 332 LOC – 76 ORG;
    - Parte di "Memorie di Monelli" (95%): 373 PER – 275 LOC – 53 ORG;
    - "Diario di Martini" (73%): 856 PER – 1029 LOC – 261 ORG;
    - Parte di "Relazione della Camera Comitati segreti" (53%): 695 PER –

474 LOC – 340 ORG.

- Test set:
    - Parte di “Discorsi di Salvemini” (24%): 5 PER – 108 LOC – 23 ORG;
    - Parte di “Memorie di Monelli” (5%): 10 PER – 25 LOC – 1 ORG:
    - Parte di “Diario di Sonnino” (27%): 264 PER – 421 LOC – 94 ORG.
    - Parte di “Relazione della Camera Comitati segreti” (47%): 560 PER – 477 LOC – 297 ORG;

Così distribuite le Named Entities costituiscono un totale di 1946 PER – 2110 LOC – 730 ORG per il training set, ovvero rispettivamente il 70% - 67% - 64% del *gold standard*, mentre per il test set un totale di 839 PER – 1031 LOC – 415 ORG, rispettivamente il 30% - 33% - 36% del *gold*. Una volta bilanciati e ripartiti i dati sono pronti per essere acquisiti come input per la progettazione del classificatore di Named Entities. Le figure 2, 3, 4 mostrano la divisione delle differenti classi di NEs tra train-set e test set.

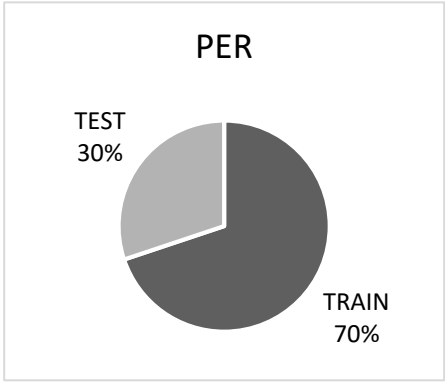

Figura 2. Suddivisione della classe PER

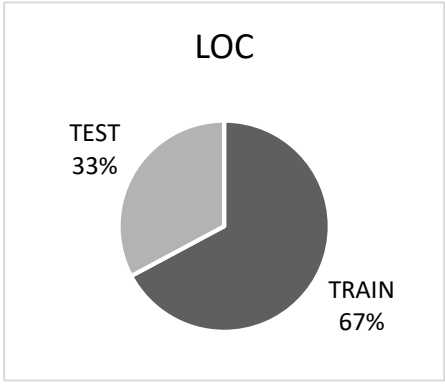

Figura 3. Suddivisione della classe LOC

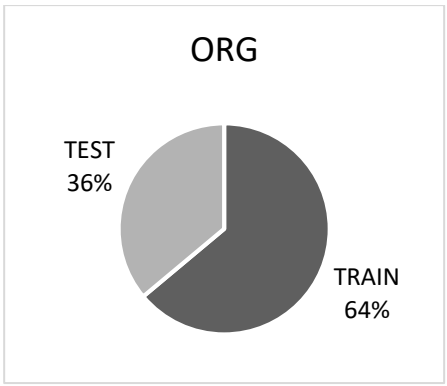

Figura 4. Suddivisione della classe ORG

Infine, per creare un addestramento più corposo e affidabile sono stati aggiunti al training set sopra menzionato i seguenti corpora annotati manualmente:

- I-CAB (Italian Content Annotation Treebank), corpus di notizie italiane composto da 525 documenti, in cui la classe "GPE" è stata convertita in "LOC" e quella "MIL" in "ORG";
- Bollettini WWI, corpus di bollettini pubblicati in "I bollettini della Guerra 1915-1918";
- Bollettini WWII, corpus di bollettini risalenti alla Seconda Guerra Mondiale, da cui sono state prese solo frasi contenenti entità militari.

La distribuzione delle entità nominate per questa parte di dati è di 4670 PER, 5920 LOC e 4098 ORG, che sommate a quelle di VGG compongono un training set finale di 7455 nomi di persona, 9061 località e 5244 organizzazioni.

Per una panoramica generale della composizione di training set e test set sono riportati di seguito i grafici che li descrivono:

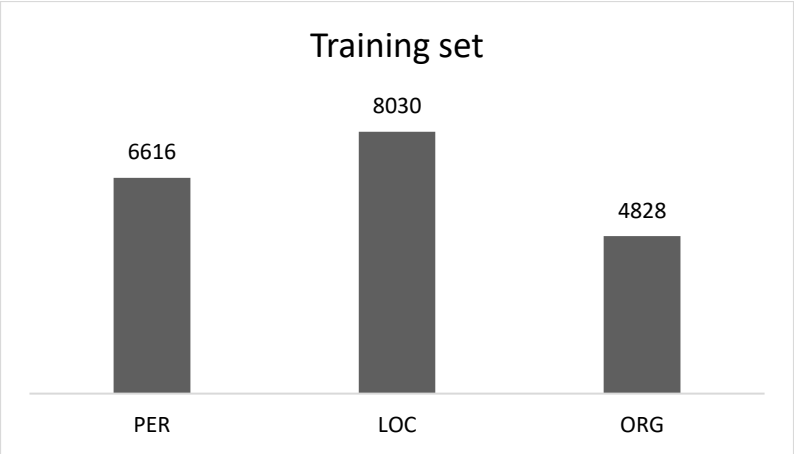

Figura 5. Distribuzione delle NEs nel training set

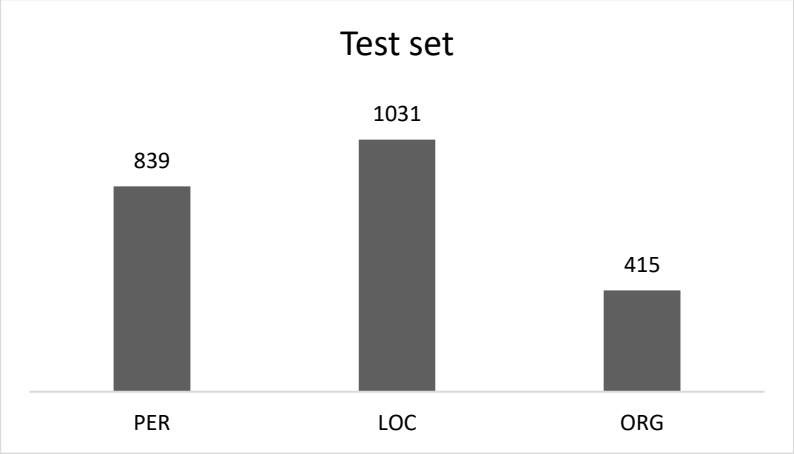

Figura 6. Distribuzione delle NEs nel test set

4.1 Gazetteer

A questo punto è opportuno creare un gazetteer, un file di testo `.lst` contenente tutte le entità estratte dai *gold standard* del training set, strutturato come un elenco di nomi (persone, luoghi e organizzazione) a cui viene associata la label corrispondente (PER, LOC, ORG). Quest'ultima rappresenta proprio la classe che il modello nel training viene addestrato a riconoscere.

Nel gazetteer ogni coppia nome-label deve essere riportato nella seguente formattazione:

```
<Tag>\t<term>\n
```

Tali dizionari non sono certamente facili da realizzare, soprattutto se i *gold standard* da cui vengono estratte le entità hanno strutture interne diverse tra loro, come nel caso di quelli creati per VGG; infatti, si è ritenuto opportuno creare due gazetteer differenti,

uno da utilizzare in fase di training e uno, più grande, per il test.
Quello che chiamiamo "train gazetteer" contiene un totale di 7884 entità (3596 PER, 2211 LOC, 2077 ORG) e sarà impiegato per l'addestramento con gazetteer del classificatore. Per quanto riguarda invece il "test gazetteer", la preparazione è stata piuttosto impegnativa:

1. è stato appositamente generato un programma python che prende in input i documenti del training set (uno alla volta) annotati nel formato CoNNL-U e restituisce un file `.lst` con l'elenco "TAG-NE" nella formattazione del gazetteer sopra citata. Ad esempio:
   ```
   LOC Albania
   ```
2. successivamente è stato normalizzato (con la stessa struttura del nuovo gazetteer) un gazetteer preesistente formato dalle seguenti entità:
   a. nomi di organizzazioni militari italiane nel corso della Prima Guerra Mondiale ricavati da Wikipedia;
   b. nomi di persone vissute nella prima metà del '900 estratti dall'enciclopedia Treccani;
   c. nomi di luogo forniti dal CNR.
3. infine è stato effettuato un merge dei due gazetteer realizzati, salvato in un file `.lst` e ripulito dei duplicati, sempre mediante apposito scrpit python.

Il risultato di questa operazione è quindi un gazetteer molto ampio comprendente 579.345 entità nominate con la seguente distribuzione: 100.782 PER, 9154 LOC, 469.409 ORG.

4.2 Training

L'addestramento dell'algoritmo supervisionato di ML rappresenta una delle forme più semplici di apprendimento induttivo automatico, in quanto il modello alla base dell'algoritmo consiste proprio in una generalizzazione ottenuta da un insieme di dati osservati, piuttosto che in un insieme di regole definite da applicare a casi specifici (Pustejovsky & Stubbs, 2013). Allo stesso modo l'addestramento del nostro VGG NER si basa esclusivamente sull'osservazione: dato un insieme iniziale di esempi (*gold standard*) su cui costruire l'algoritmo decisionale, il classificatore ne apprende le strutture grammaticali sulla base di features particolarmente rilevanti ed elabora delle ipotesi per ricostruire la regola generale (funzione obiettivo) di tagging fino a

quel momento sconosciuta; è grazie ad essa che il modello addestrato è in grado di identificare e classificare correttamente le entità mai incontrate prima per annotarle con una delle classi possibili tra PER, LOC e ORG.

Nel nostro caso specifico del classificatore di NEs, l'addestramento verrà fatto per due modelli: il primo, strettamente statistico, prende in input solo i *gold standard* dei documenti sopra menzionati (v. Sezione 4), mentre nel secondo viene utilizzato anche il gazetteer, dal quale il NER apprenderà le features in base alle parole contenute in esso. L'utilità di combinare i due approcci risiede nell'opportunità di generare un modello più affidabile e con migliori prestazioni, dati i vantaggi e gli svantaggi di entrambi. Infatti, se gli approcci basati su gazetteer raggiungono risultati migliori per domini specifici, quelli unitamente stocastici basati sull'apprendimento automatico, sono più efficienti su domini diversi e sono in grado, inoltre, di eseguire analisi predittive su entità non presenti nel dizionario (gazetteer), grazie all'identificazione di alcune features; tuttavia, questi ultimi approcci richiedono grandi quantità di dati di addestramento, purtroppo non sempre disponibili.

Il gazetteer rappresenta dunque, in fase di training, un contributo fondamentale per la risoluzione di tutti quei casi di dipendenze non locali, che possono essere elencate per fare riferimenti incrociati tra i vari nomi che indicano la stessa entità, mentre l'apprendimento automatico basato sulle sequenze di parole garantisce una più semplice disambiguazione delle NEs, tenendo conto del contesto in cui si trovano.

Analizziamo ora meglio la preparazione degli strumenti necessari all'addestramento e le fasi di cui si esso si compone.

4.2.1 Stanford NER

Il primo requisito indispensabile per intraprendere la progettazione è possedere una tra le versioni dello *Stanford NER*, anche conosciuto come CRFClassifier; si tratta di una serie di strumenti di NLP, che sono complessivamente denominati *Stanford CoreNLP*, messi a disposizione dal gruppo di ricerca di Stanford University. Il software fornisce un'implementazione Java per la creazione dei modelli di sequenza CRF (Conditional Random Field) a catena lineare, nonché sequenze di processi di elaborazione per l'annotazione di documenti. Dunque, per poter eseguire Stanford NER da terminale e, quindi, avviare il training del classificatore, è necessario installare una delle versioni Java successive o uguali alla 1.8.

I modelli utilizzati da Stanford per l'addestramento si servono di caratteristiche di

similarità distribuzionale, che offrono un notevole aumento delle prestazioni, a discapito della memoria.

La parte di riga di comando per l'avvio del training, che invoca Java e i file Stanford è la seguente:

```
java -Xmx4G -cp stanford-ner.jar
edu.stanford.nlp.ie.crf.CRFClassifier
```

In particolare, il parametro "4G" specifica la porzione di RAM che verrà impiegata per l'esecuzione dell'algoritmo, da "cp" in poi invece viene specificato che si vogliono caricare i modelli del file jar dal percorso

```
edu/stanford/nlp/models …
```

4.2.2 Train prop

A questo punto occorre preparare il file di prop, una lista di parametri e proprietà che specificano quali features esaminare, da fornire alla macchina affinché svolga correttemente il task di riconoscimento; esso ha una struttura di questo tipo (la spiegazione per ogni riga è specificata dal carrattere "#"):

- **trainFileList=<path>** #path alla cartella contenente i file di train (training set)
- **serializeTo=<path>** #path al file in cui si vuole salvare (serialize) il modello risultante (nel formato `.ser.gz`[1])
- **map=** #vanno specificate le colonne del training set
- **useClassFeature=true** #specifica le feature che vogliamo addestrare
- **useTags=true** #fornisce le features per "usePrev" e "useNext"
- **useWord=true** #restituisce la feature corrispondente alla parola
- **useNGrams=true** #crea le features dagli n-grammi, le sottostringhe della parola
- **noMidNGrams=true** #non include le features

[1] Aggiungendo `.gz` alla fine il file viene automaticamente gzippato, rendendolo più piccolo e veloce per il caricamento del modello.

"character n-gram" per gli n-grammi che non contengono né l'inizio né la fine della parola

- **maxNGramLeng=6** #vengono inclusi nel modello solo n-grammi di dimensione 6
- **usePrev=true** #abilita l'uso di features precedenti
- **useNext=true** #abilita l'uso di features successive
- **useSequences=true** #abilita la combinazione delle classi
- **usePrevSequences=true** #abilita la combinazione di classi, usando le classi precedenti
- **maxLeft=3** #specifica l'ordine del CRF
- **useTypeSeqs=true** #utilizza le features di forma della parola di ordine 0
- **useTypeSeqs2=true** #aggiunge ulteriori features di forma della parola di ordine 1 e 2
- **useTypeySequences=true** #usa alcuni modelli di forma di parola di ordine 1
- **wordShape=chris2useLC** #specifica il nome di una funzione di forma di parola
- **useDisjunctive=true** #include le features che generano disgiunzioni di parola

Occorre fare alcune precisazioni sul campo "map": qui vanno inserite tutte le colonne separate da tab nel file .conll (*gold standard*), di cui è necessario specificare con l'attributo "word" la colonna corrispondente alla parola in input del nostro training set, mentre con "answer" quella corrispondente all'output.

Il file di prop per il modello con gazetteer è strutturato allo stesso modo, ma con l'aggiunta dei seguenti parametri:

- **cleanGazette=true** #una feature nel gazetteer si attiva solo quando si incontra l'esatta corrispondente nel testo di train
- **useGazettes=true** #usa le features del gazetteer
- **gazette= <path>** #path al gazetteer

Il file di prop viene così salvato e aggiunto al comando menzionato precedentemente (v. Sezione 4.2.1), che verrà eseguito all'interno di una shell per la costruzione del classificatore:

```
java -Xmx4G -cp stanford-ner.jar
edu.stanford.nlp.ie.crf.CRFClassifier -prop
<percorso_file_props>
```

4.2.3 Elaborazione del codice e addestramento

Nel processo di addestramento basato sull'apprendimento automatico, il training-set si presenta come un insieme di coppie nella forma (input ; output) $\Rightarrow$ (x_1 ; y_1), (x_2 ; y_2), …, (x_n ; y_n), in cui l'input x_i è detto evento e corrisponde al token, mentre l'output y_i è una tra le possibili classi ammesse come soluzione del problema da risolvere, quindi assegnabili al token (PER, LOC, ORG). In generale, nei problemi affrontati con algoritmi supervisionati l'insieme delle possibili classi di output deve essere finito, mentre lo è raramente quello degli eventi in input. Dunque, scopo della fase di addestramento è l'apprendimento di una nuova funzione $f(x_i)$, detta funzione obiettivo, che lega la variabile determinante x (il token) alla variabile determinata y (la classe), ovvero tale che data una serie di classi y_1, …, $y_n \Rightarrow f(x_i) = y_i$.

Per prima cosa vengono selezionate delle caratteristiche salienti, dette features, all'interno del corpus annotato, determinando il grado di accuratezza del sistema finale. Nei task di NLP le features possono essere locali, contestuali e globali: quelle globali sono estratte da contesti più ampi rispetto a quelle contestuali (ad esempio il dominio testuale), che invece vengono estratte direttamente dal contesto in cui il token in analisi si trova, quindi ad esempio la parola precedente, quella successiva, la POS della parola precendente, la POS di quella successiva, ecc.; le features locali sono quelle che vengono estratte dal token in analisi, quindi forma, lemma, suffisso, prefisso, ecc. Dunque, dopo essere state scelte, le features vengono estratte da tutto il corpus, processo che restituisce per ogni coppia (input, output), la lista delle features attive in quel contesto per la classe output. A questo punto l'algoritmo di apprendimento supervisionato calcola la frequenza delle features e assegna loro un peso, che rappresenta la salienza che hanno nell'indicare una certa classe come possibile output.

Per il nostro esperimento abbiamo addestrato il modello con le features usate dal NER di Stanford:

- Features morfologiche e ortografiche, che tengono conto sia delle parole precedenti e successive nelle sequenze (ad esempio in “generale Cadorna”, la parola “generale” aiuta a classificare l’entità che la segue come PER), che della presenza di elementi particolari come lettere maiuscole, caratteri non alfabetici, ecc.;
- Features linguistiche, quali la posizione della parola nella frase, il lemma e il PoStag;
- Termini complessi, che permettono di considerare gruppi di parole come una singola entità (ad esempio “Ministro degli Esteri”).

Quindi, sommariamente, il nostro classificatore altro non fa che estrarre dal training set le features, calcolare la distribuzione di frequenza tra queste e gli output a esse associati e utilizzarla per calcolare i parametri della funzione obiettivo stimata per la classificazione delle entità nel testo. In altri termini, il NER in fase di addestramento è in grado di apprendere le varie caratteristiche proprie di un determinato elemento linguistico appartenente ad una classe e di valutare per ognuna di queste features il contributo (peso) nel determinare la corrispondente classe di appartenenza (Dell’Orletta F. e Venturi G., 2016).

Per quanto riguarda l’approccio con gazetteer invece il procedimento di addestramento del NER è lo stesso, con l’unica differenza che esso apprende le features anche in base alle parole contenute nel dizionario: data una certa entità nel testo, il peso che il classificatore associa alla classe da assegnare è maggiore se l’entità è citata nel gazetteer con quella stessa classe. Tuttavia, va specificato che, l’utilizzo del gazetteer nel training non garantisce che le sue voci vengano sempre utilizzate come membri della classe prevista, né tantomeno che le parole al di fuori di esso non vengano scelte, in quanto fornisce semplicemente una feature aggiuntiva per cui il classificatore deve addestrarsi, ma se quest’ultimo ha pesi maggiori per altre features, quelle del gazetteer possono essere ignorate.

4.3 Test

Terminata la fase di addestramento dei due modelli, con quella di test (analisi) ci proponiamo di valutare le prestazioni del modello creato, per verificare l’affidabilità del nostro sistema di Named Entity Recognition. Come sopra descritto (v. Sezione 4), il test set utilizzato ha la stessa struttura dell’insieme di training, ma un campione di dati differenti, sia in termini di contenuti che di quantità, in quanto rappresenta soltanto

il 30% circa del nostro GS totale (mentre il training set ne costituiva circa il 70%). La necessità di fornire in fase di test un nuovo set di dati, nasce dall'esigenza di valutare proprio la capacità dell'algoritmo di generalizzare quanto appreso nel training.
Inoltre, come già precisato (v. Sezione 4.1), anche per il processo di analisi viene fornito un gazetteer, le cui entità saranno trattate semplicemente come informazioni aggiuntive da consultare per l'assegnamento delle classi alle entità.

4.3.1 Stanford NER

Anche per valutare il modello è opportuno servirsi di alcuni moduli forniti dallo standard Stanford; in particolare, tramite la classe *NERClassifierCombiner* è possibile accedere ad una funzionalità della pipeline *Stanford CoreNLP*, che permette di utilizzare più CRF insieme. Nel nostro caso andiamo a combinare nel test i due modelli creati dal training, quello con Gazetteer, e quello senza, specificandoli in seguito al comando `-ner.model` nel codice che eseguiremo da shell (medesimo nella parte di Java a quello usato per il training) per l'avvio del processo. Inoltre, per il nostro scopo occorre disattivare alcune delle opzioni previste dal modulo *NERClassifierCombiner*, impostandole al valore booleano "false"; si tratta di `ner.applyNumericClassifiers` e `ner.useSUTime` che, se attive, riconoscono e codificano sequenze numeriche e sequenze correlate al tempo, producendo quindi tag come NUMBER, ORDINAL, MONEY, DATE e TIME, che non rientrano nel tagset definito per questa progettazione.
Dunque, il comando lanciato da shell per eseguire il test ha la seguente struttura:

```
java -Xmx4g -cp stanford-ner.jar
edu.stanford.nlp.ie.NERClassifierCombiner -prop
<percorso_file_props> -ner.model
<percorso_file_modelloSENZAgaz>,
<percorso_file_modelloCONgaz>
-ner.applyNumericClassifiers false -ner.useSUTime false
> <percorso_file.results>
```

Spieghiamo nella sezione successiva il significato e il contenuto del file props; mentre `file.results` rappresenta il file di output con il corpus di test annotato in cui la prima colonna corrisponde al token di input, la seconda alla risposta corretta (*gold*) e la terza alla risposta predetta dal classificatore.

4.3.2 Test prop

Anche per la fase di test è necessario preparare un file di prop in cui specificare le proprietà del modello che si vogliono prendere in analisi; esso ha la seguente struttura (simile a quella del `train.props`):

- **`testFile=<path>`** `#path alla cartella contenente i file di test (test set)`
- **`map=`** `#vanno specificate le colonne del test set`
- **`cleanGazette=true`** `#una feature nel gazetteer si attiva solo quando si incontra l'esatta corrispondente nel testo di test`
- **`useGazettes=true`** `#usa le features del gazetteer`
- **`gazette=<path>`** `#path al gazetteer`

La voce "map" viene compilata con lo stesso criterio e le stesse diciture di quella presente nel `train.props`.

4.3.3 Elaborazione e analisi

Nel processo di analisi del modello ogni token da analizzare viene rappresentato come un vettore booleano di features attive per esso, in cui lo 0 indica che la feature non è attiva per il token, mentre l'1 che lo è. A ogni vettore X è associata la variabile decisionale Y, ossia la classe di assegnamento corretta (del *gold*). Un numero n di vettori, che costituiscono il campione di dati su cui testare il classificatore, vengono presi in input dalla macchina, la quale assegna un punteggio a ogni possibile classe di output (PER, LOC, ORG); tale punteggio viene calcolato sulla base delle features estratte dal testo da analizzare, utilizzando i pesi di quelle apprese dall'algoritmo in training. Dunque, la classe a cui viene assegnato lo score più alto è la classe vincente, nonché il risultato dell'analisi, che viene confrontato a questo punto con la risposta corretta per valutare la performance del sistema nei task di riconoscimento e classificazione delle entità. Se la decisione della macchina coincide con quella del *gold standard* il nostro Named Entity Recognizer ha superato il test, in caso contrario vanno individuati gli errori che commette ed è necessario riaddestrare il modello fino a che non produce risultati soddisfacenti.

Quindi, sommariamente, in fase di analisi, il NER è in grado di estrarre da un nuovo corpus annotato le features che contribuiscono a determinare la classe di appartenenza

di un determinato elemento, verificare la presenza di quest'ultimo nel gazetteer fornito ed eventualmente assegnargliela.

5 Analisi dei risultati

Per valutare al meglio quanto effettivamente il NER addestrato ha migliorato le sue prestazioni nel riconoscimento delle entità nominate all'interno del corpus VGG, lo confrontiamo ora con quello creato per un progetto molto simile dal nome *Memorie di Guerra,* e con un primo prototipo che era stato utilizzato per VGG, entrambi testati sullo stesso set di dati di VGG.

Per il progetto *Memorie di Guerra* sono stati annotati i bollettini emessi dal Comando Supremo con il resoconto quotidiano delle operazioni nella Grande Guerra e nella Seconda guerra Mondiale ed un primo classificatore è stato realizzato adattando un NER preesistente, addestrato su I-CAB, ai Bollettini; quest'ultimo è stato poi soggetto a un esperimento intermedio che lo ha adattato ai testi di VGG, ma selezionati in quantità ridotta e senza un rigoroso bilanciamento nella suddivisione dei generi testuali dell'intero corpus, né in quella delle entità tra training e test set. È quindi con il nostro VGG NER, che conta su una preparazione degli strumenti necessari per la progettazione più mirata e precisa (v. Sezione 4) ed è stato addestrato su un corpus di testi abbastanza ampio, rappresentativo della varietà linguistica e del genere testuale, che poniamo la fiducia in un miglioramento ulteriore delle performance di annotazione.

Inoltre, è bene specificare che se pur i corpora impiegati per l'addestramento dei tre classificatori sono molto diversi tra loro, il set di features utilizzate è lo stesso (v. Sezione 4.2.3).

Quindi, riepilogando, analizzeremo i seguenti modelli:

1) **I-CAB / *Memorie di guerra***
    - Training
        a) training set: I-CAB + Bollettini
        b) gazetteer: entità estratte dal training set
    - Test
        a) test set: VGG
        b) gazetteer: entità estratte dal training set + VGG
2) **I-CAB / *Memorie di Guerra* / VGG (ridotto)**
    - Training
        a) training set: I-CAB + Bollettini + VGG
        b) gazetteer: entità estratte dal training set

- Test
    a) test set: VGG
    b) gazetteer: entità estratte dal training set + VGG

3) I-CAB / Memorie di Guerra / VGG (completo)

- Training
    a) training set: I-CAB + Bollettini + VGG
    b) gazetteer: entità estratte dal training set
- Test
    a) test set: VGG
    b) gazetteer: entità estratte dal training set + VGG

5.1 Metriche di valutazione

Alla fine del processo di analisi (test) vengono mostrati in schermo i risultati ottenuti, che consistono nei valori assegnati ad alcune metriche di valutazione dell'accuratezza del sistema nell'etichettere il set di dati: *Precision* (P), *Recall* (R), *F-measure* (F1), true positive (TP), false positive (FP) e false negative (FN).

- *Precision* misura il grado di correttezza delle risposte date dal classificatore, calcolato sul rapporto tra gli output che ha etichettato correttamente e quelli che ha restituito in totale;
- *Recall* misura la copertura del sistema, cioè prende in considerazione anche i tag corretti che il NER si è perso o che ha sostituito con tag errati, quindi il rapporto tra gli output corretti e quelli che avrebbe dovuto restituire;
- *F-measure* rappresenta la combinazione dei valori di P e R nella loro media armonica;
- "true positive" sono gli output corretti, il numero degli output del sistema attestati nel GS;
- "false positive" sono gli output errati, il numero degli output trovati dal sistema non attestati nel GS;
- "false negative" sono gli output mancanti, il numero dei casi nel GS per i quali il sistema non produce un output o produce un output errato.

Dunque, *Precision*, *Recall* ed *F-measure* sono calcolate come segue:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F1 = \frac{2\,PR}{P + R}$$

In genere P e R sono valori in competizione: aumentare la precisione del sistema porta spesso a penalizzare la *Recall* e viceversa.

5.2 Modello I-CAB e Bollettini

Nel primo esperimento sono state annotate le entità dei testi di VGG (del test set) con il NER di *Memorie di Guerra*, addestrato sulla combinazione di I-CAB, corpus di notizie italiane esponente del linguaggio standard contemporaneo, con i Bollettini della Grande Guerra, che riflettono l'italiano arcaico del primo Novecento.
La Tabella 1 mostra i risultati del test:

Entity	P	R	F1	TP	FP	FN
B-LOC	0,8353	0,8272	0,8312	852	168	178
B-ORG	0,4955	0,3942	0,4391	164	167	252
B-PER	0,8977	0,4702	0,6172	395	45	445
Totals	0,7878	0,6172	0,6922	1411	380	875

Tabella 1. Risultati test su modello I-CAB + *Memorie di guerra*

Come si può ben notare e come ci si aspettava, il NER non ha ottenuto grandi risultati sul corpus VGG, soprattutto nell'identificazione di nomi organizzazione e di persona. Ciò è dovuto chiaramente alla differenza tra il training e il test corpus, sia nella lingua, sia nella varietà dei generi testuali. Infatti, come già specificato nelle sezioni precedenti, se I-CAB è un campione dell'italiano standard contemporaneo e i testi di *Memorie di Guerra* coprono un dominio testuale molto specifico, riservato appunto ai Bollettini di guerra, *Voci della Grande Guerra*, presenta invece una certa eterogeneità sia nel genere che nel registro linguistico. Sicuramente, anche la diversa distribuzione di NEs, che in VGG sono leggermente sbilanciate verso le località, ha contribuito a registrare per queste i più alti valori di *Precision*, *Recall* ed *F-Measure*.

5.3 Modelli con VGG

Passiamo ora ad analizzare i due modelli che si sono serviti dei testi di VGG sia, in parte, in fase di addestramento, che totalmente in fase di analisi.

La Tabella 2 mostra come si comporta il NER non bilanciato e ridotto addestrato su I-CAB, Bollettini di *Memorie di Guerra* e VGG, nell'annotazione di VGG:

Entity	P	R	F1	TP	FP	FN
B-LOC	0,9372	0,9272	0,9322	955	64	75
B-ORG	0,7774	0,4952	0,6050	206	59	210
B-PER	0,9394	0,7571	0,8385	636	41	204
Totals	0,9164	0,7861	0,8462	1797	164	489

Tabella 2. Risultati test su modello I-CAB + *Memorie di guerra* + VGG (ridotto)

Da una prima analisi dei valori di *F-Measure* emerge che in questo esperimento il NER ha prodotto risultati più alti e soddisfacenti rispetto al precedente per tutte e tre le classi (PER, LOC, ORG) . L'aggiunta dei testi di VGG, per quanto la loro selezione sia stata casuale e non rigorosa, ci ha permesso di migliorare le prestazioni totali del classificatore del 15% e di ottenere l'incremento maggiore proprio per le classi che nel test con I-CAB e Bollettini avevano ottenuto i valori più bassi: ORG +16% e PER +22%.

Riportiamo, infine, nella Tabella 3 i risultati che ha prodotto il nostro VGG NER addestrato, e testato sul minuzioso bilanciamento di training e test set.

Entity	P	R	F1	TP	FP	FN
B-LOC	0,9477	0,9477	0,9477	978	54	54
B-ORG	0,8702	0,7590	0,8108	315	47	100
B-PER	0,9420	0,8510	0,8942	714	44	125
Totals	0,9326	0,8780	0,9045	2007	145	279

Tabella 3. Risultati test su modello I-CAB + *Memorie di guerra* + VGG (completo)

Da una prima osservazione sembrerebbe emergere che i valori riportati sono anche questa volta significativamente più alti; infatti, il nuovo modello addestrato (VGG NER) ci ha permesso, in termini di *F-Measure*, di aumentare le prestazioni totali del 6%. Se ci focalizziamo sulle singole classi, vale la pena notare il considerevole miglioramento delle organizzazioni (+21%), rispetto alle altre due categorie, PER

(+6%) e LOC (+2%), che invece hanno prodotto, in media, risultati più vicini a quelli dell'esperimento precedente.

5.4 Analisi degli errori e valutazione dei risultati

Confrontando e analizzando i tre modelli di NER proposti ci si accorge subito del netto miglioramento registrato da quello realizzato per questo lavoro (VGG NER) rispetto al primo (I-CAB + Bollettini) pari per l'esattezza al 21% in termini di *F-Measure*.

In particolare, se ci si sofferma sui risultati ottenuti per la classe ORG, come già accennato, emerge che questa rappresenta proprio la classe più interessata dal miglioramento del classificatore (+37%). È importante focalizzare l'attenzione su questa classe poiché ci si può rendere conto che l'ottimizzazione nel riconoscimento delle organizzazioni riflette esattamente la bontà del nostro metodo di combinare i due approcci, con gazetteer e di apprendimento automatico nella fase di training, e di fornire un gazetteer più ampio, in fase di test; infatti, se sia nel training set, che nel test set, che nel "train gazetteer" si contano un numero di nomi di organizzazione inferiore rispetto a persone e località (v. Figure 5 e 6), è invece nel "test gazetteer" che esse sono presenti in quantità di gran lunga maggiori, permettendo quindi al nostro NER di riconoscerle nel testo in maniera più rigorosa. Questo fatto indica anche che, sebbene il numero di dati forniti in addestramento per la classe ORG sia quello minore tra i tre, il classificatore addestrato sfrutta molto bene l'apprendimento automatico di features per riconoscere ed etichettare le organizzazioni, dunque, ha un'alta capacità di generalizzazione. Inoltre, si può definire per il VGG NER totalmente superato l'adattamento al linguaggio arcaico, di guerra, del primo Novecento.

Un'ulteriore dimostrazione di quanto esposto circa l'interpretazione dei risultati per la classe ORG è presentata dai valori di distribuzione di frequenza di alcuni esempi riportati nella Tabella 3, che evidenziano il significativo miglioramento del VGG NER proprio nel riconoscimento di nomi di organizzazioni prettamente di tipo politico-militari:

- la prima colonna riporta l'entità;
- la seconda colonna conta il numero di volte in cui l'entità compare nel test set;
- la terza colonna conta la frequenza con cui il NER "I-CAB + Bollettini" non ha riconosciuto e annotato l'entità come "ORG" nel test set;

- la quarta colonna indica quante volte il VGG NER non ha riconosciuto e annotato l'entità con la classe "ORG" nel test set.

NE	Test corpus	I-CAB + Bollettini	VGG NER
Comitato segreto	30	25	0
Conferenza di Londra	2	2	0
Consiglio	17	8	0
Impero	14	14	1
Marina	10	10	2
Ministero	17	13	2
Parlamento	27	16	0
Patto di Londra	5	5	0
Polizia militare	6	6	0
Trattato della Triplice	5	5	0

Tabella 4. Esempi annotazione FN

Dunque, sia dal notevole divario nei valori di FN mostrati dalle tabelle che descrivono le prestazioni dei due modelli (v. Tabelle 1 e 3), sia da quest'ultima analisi più approfondita degli errori d'annotazione nel corpus, si può intuire ancora una volta il ruolo fondamentale che rivestono le organizzazioni di tipo politico-militare nel corpus VGG e, più in generale, nei testi di guerra e l'accuratezza del nostro classificatore nell'individuarle.

Infine, un breve confronto con il NER per il cui addestramento sono stati impiegati soltanto una minima parte di testi del corpus VGG (v. Tabella 2) e senza porre particolare attenzione nella selezione di questi al bilanciamento nel genere testuale e nella partizione tra training set e test set, ci permette di osservare un miglioramento ulteriore del nostro classificatore. Dunque, ne consegue che, definire in maniera particolarmente rigorosa, sia qualitativamente che quantitativamente, i dati da fornire in fase di addestramento è fondamentale per raggiungere risultati ottimali nei task di Information Extraction come quello di Named Entity Recognition.

Per poter confrontare meglio il progressivo miglioramento riscontrato per l'annotazione delle entità in VGG dal NER realizzato per questo lavoro rispetto a quelli precedenti, sono riportati di seguito i grafici che li comparano in termini di *Precision, Recall* e *F-Measure*:

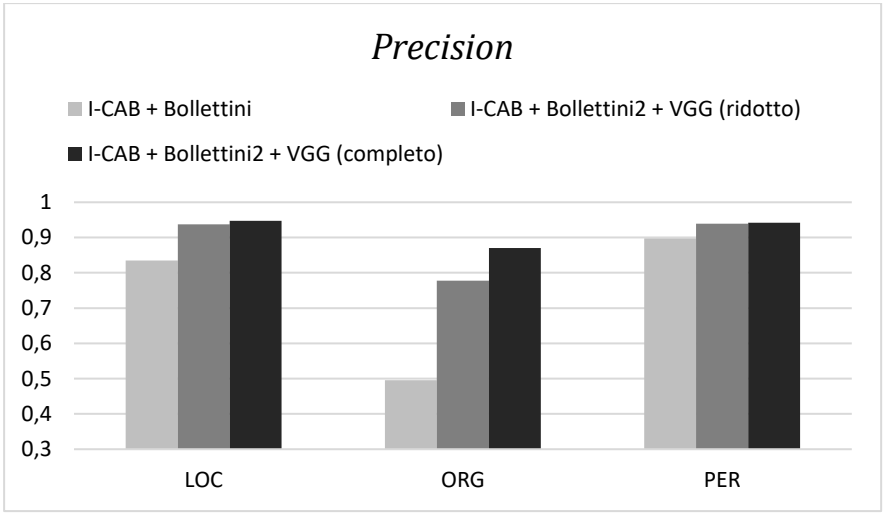

Figura 7. Confronto delle classi dei tre modelli per valori di *Precision*

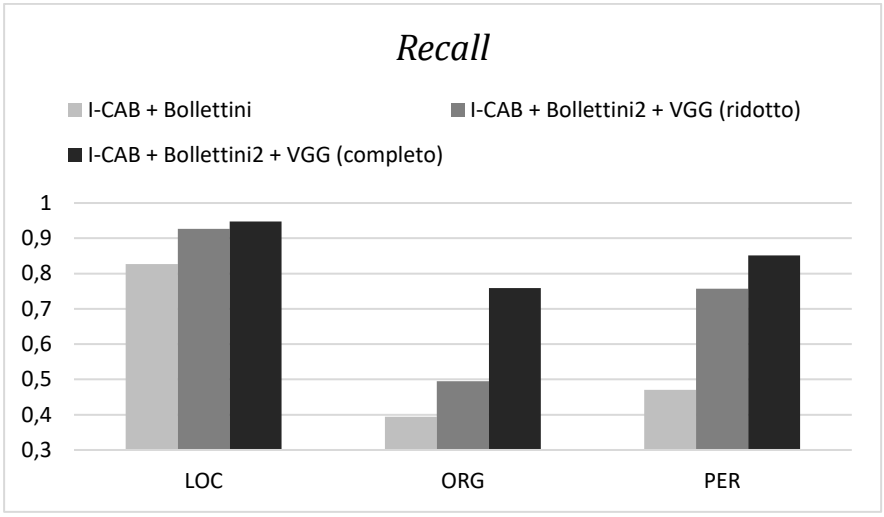

Figura 8. Confronto delle classi dei tre modelli per valori di *Recall*

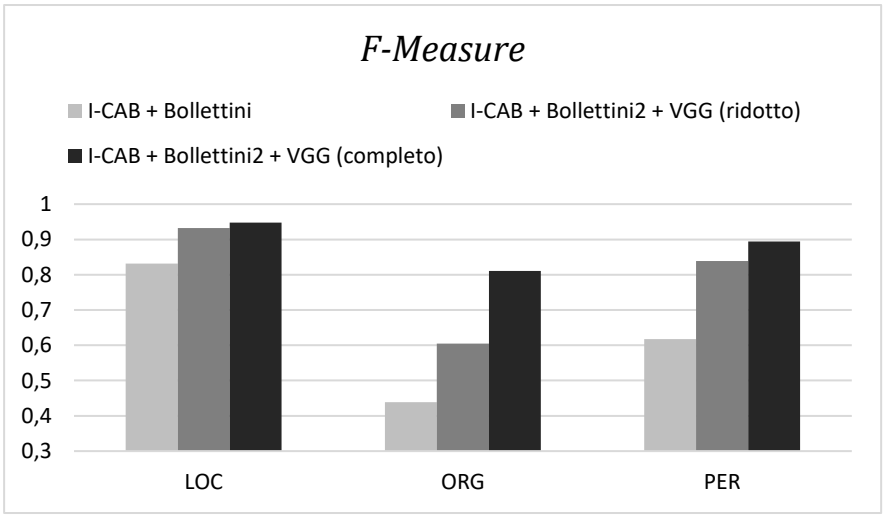

Figura 9. Confronto delle classi dei tre modelli per valori di *F-Measure*

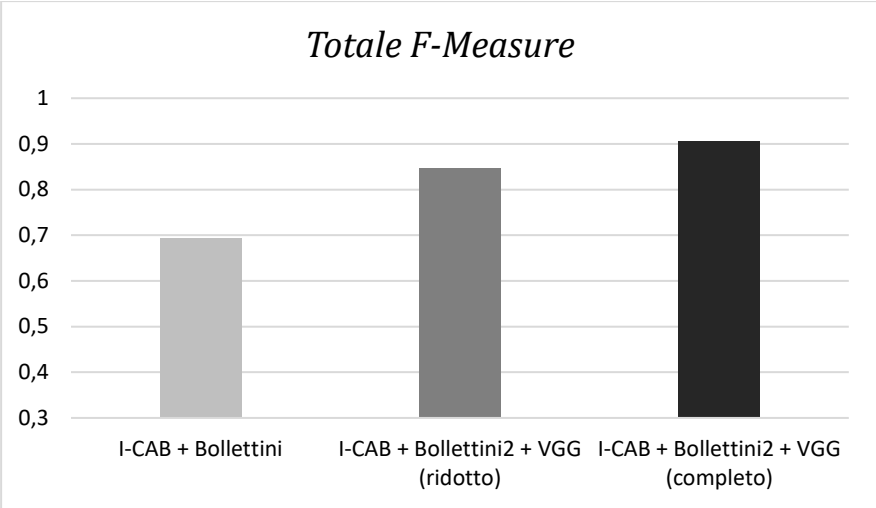

Figura 10. Confronto dei tre modelli per valori di F-Measure totali

6 Conclusioni

Il riconoscimento e classificazione delle entità nominate nei testi è uno dei task principali dell'Information Extraction in NLP e questo lavoro si propone di risolverlo progettando un NER basato sull'apprendimento automatico (Machine Learning) per adattarlo ai testi della Prima Guerra Mondiale messi a disposizione dal progetto *Voci della Grande Guerra*.

Dopo una breve rassegna delle difficoltà e delle sfide affrontate per l'addestramento del classificatore, l'attenzione si è spostata sui risultati prodotti in fase di test, nonché sulla valutazione dell'efficienza del lavoro svolto, dunque del modello creato. Sulla base di quest'ultima analisi affrontata e del confronto con i NER preesistenti, il progressivo miglioramento registrato nell'annotazione di testi della Grande Guerra dal VGG NER ci permette di ritenere raggiunto, con ottimi risultati, il nostro obiettivo di Domain adaptation a testi del primo Novecento eterogenei nel genere testuale (discorsi, diari, memorie, relazioni), nel registro linguistico (arcaico colto, arcaico popolare, ufficiale militare, politico, giuridico, ecc.) e nei temi affrontati (guerra, politica, quotidianità, ecc.).

Inoltre, il particolare successo conseguito dal VGG NER nel riconoscimento della classe ORG, ha permesso di realizzare anche uno degli obiettivi finali del progetto VGG e più in generale della Linguistica Computazionale: l'estrazione di informazioni (IE). Dunque, risulta evidente l'importanza del ruolo che rivestono le organizzazioni, specie quelle militari e politiche, nei testi storico-politici e, quindi, nella guerra; a confermarlo è proprio l' analisi dell'annotazione.

Nonostante la Grande Guerra sia stata un punto di riferimento fondamentale nella storia dell'Umanità, ad oggi la memoria di un evento così importante sta progressivamente scomparendo tra le nuove generazioni, cosìcchè affrontare le sfide del Natural Language Processing per adattare gli attuali strumenti lingustici a testi del primo Novecento, rappresenta un ottimo punto di partenza per l'impiego delle memorie di pietre miliari del passato come fonte per le nuove scoperte del futuro, oltre che riflettere perfettamente l'emergente intersezione delle Digital Humanities con la Linguistica Computazionale.

6.1 Nuove frontiere

Negli ultimi anni è stato dimostrato che architetture come le reti neurali raggiungono

prestazioni allo stato dell'arte per una vasta gamma di attività del NLP, tra cui proprio quella di codifica di sequenze di parole (Named Entity Recognition).

Uno dei principali punti di forza delle reti neurali rispetto agli altri sistemi del NLP, consiste nella considerevole abilità che queste hanno proprio nel rilevare automaticamente le features di carattere, registrando per molti esperimenti valori di F1 più alti del 90%. Per il task di NER, uno dei più qualificati a livello di performance dei risultati è il modello *Bi-directional LSTM*, una rete bidirezionale in grado di tenere conto dei contesti su entrambi i lati (destro e sinistro) di una parola, risolvendo la maggior parte dei problemi di dipendenze locali delle entità e di disambiguazione nel testo (Graveset al., 2013).

Come abbiamo visto il Named Entity Recognition è un compito di apprendimento impegnativo e, sfortunatamente, risorse e features linguistiche specifiche sono costose da sviluppare in nuove lingue e nuovi domini, rendendo il NER una sfida da adattare. Tuttavia, testando un gran numero di configurazioni della rete neurale per il NER è stato dimostrato che alcune scelte conducano a prestazioni molto alte e dipendono meno dalla restante configurazione della rete, così da richiedere meno adattamente quando vengono applicate a nuove lingue o domini.

7 Bibliografia

- Alessandro Lenci, Nicola Labanca, Claudio Marazzini, Simonetta Montemagni. 2016. *Voci della Grande Guerra: An Annotated Corpus of Italian Texts on World War I*. "Italian Journal of Computational Linguistics", pp. 101–108.
- Irene De Felice, Felice Dell'Orletta, Giulia Venturi, Alessandro Lenci, Simonetta Montemagni. 2018. *Italian in the Trenches: Linguistic Annotation and Analysis of Texts of the Great War*. "Proceedings of 5th Italian Conference on Computational Linguistics (CLiC-it), 10-12 Dicembre, 2018", pp. 160-164.
- Lenci A., Labanca N., Marazzini C., Montemagni S., Boschetti F., De Felice I., Dei Rossi S., Dell'Orletta F., Di Giorgio M., Passaro L., Venturi G. 2018. *Voci della Grande Guerra Preserving the Digital Memory of World War I*. "Proceedings of 7th AIUCD Annual Conference, 31 Gennaio – 2 Febbraio, 2018", pp. 196- 197.
- Lucia C. Passaro, Alessandro Lenci. 2014. "*Il Piave mormorava…": Recognizing Locations and other Named Entities in Italian Texts on the Great War. "*Proceedings of the First Italian Conference on Computational Linguistics, Pisa, Italy, Dicembre 9-10 2014", pp. 286-290.
- Jerry R. Hobbs, Ellen Riloff. 2010. *Information Extraction*. "Handbook of Natural Language Processing", pp.16-18.
- Daniel Jurafsky & James H. Martin. 2018. *Speech And Language Processing: An Introduction to Natural Language Processing , Computational Linguistics, and Speech Recognition*. Third Edition draft.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*. "Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)", pp. 363-370.
- Gallerani Roberto. 2015. *Natural Language Processing (NLP) e Information Extraction (IE).* [ebook] Bologna: APPUNTI DIGITALI Quaderni. Available at: https://www.gallerani.it/wordpress/wp-content/uploads/2015/12/Natural-Language-Processing-e-Information-Extraction-1.pdf
- J. Pustejovsky & A. Stubbs. 2012. *Natural Language Annotation for Machine Learning*. Sebastopol, O'Reilly Media.
- Cristopher D. Manning, Prabhakar Raghavan & Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
- Bengfort, B. 2012. *A Survey of Stochastic and Gazetteer Based Approaches for Named Entity Recognition.*
- Dell'Orletta F., Venturi G. 2016. *ULISSE: una strategia di adattamento al dominio per l'annotazione sintattica automatica*. "E. M. Ponti e M. Baudassi (a cura di) *Compter parler soigner: tra linguistica e intelligenza artificiale"*, atti del convegno 15-17 dicembre 2014, Pavia University Press.
- Nils Reimers & Iryna Gurevych. 2017. *Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging*. "Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing", pp. 338–348.
- Chiu J.P. & Nichols E. 2016. *Named Entity Recognition with Bidirectional LSTM-CNNs*. Transactions of the Association for Computational Linguistics, 4, pp. 357-370.
- Naji F. Mohammed & Nazlia Omar. 2012. *Arabic Named Entity Recognition Using Artificial Neural Network.* Journal of Computer Science, vol. 8, pp. 1285-1293.

8 Sitografia

- http://www.vocidellagrandeguerra.it/
- https://universaldependencies.org/format.html
- https://nlp.stanford.edu/software/CRF-NER.html
- https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html
- https://stanfordnlp.github.io/CoreNLP/ner.html#fine-grained-ner
- http://www.memoriediguerra.it/site/page?view=about