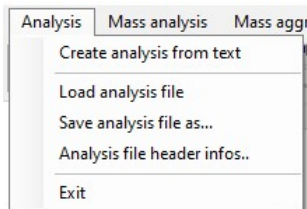
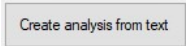


TextAnalyzer 010 014 - Manuale

Menù analisi



Analisi di un singolo testo

“Create analysis from text” permette di scegliere ed analizzare un file in formato .txt (è disponibile anche dal pulsante ).

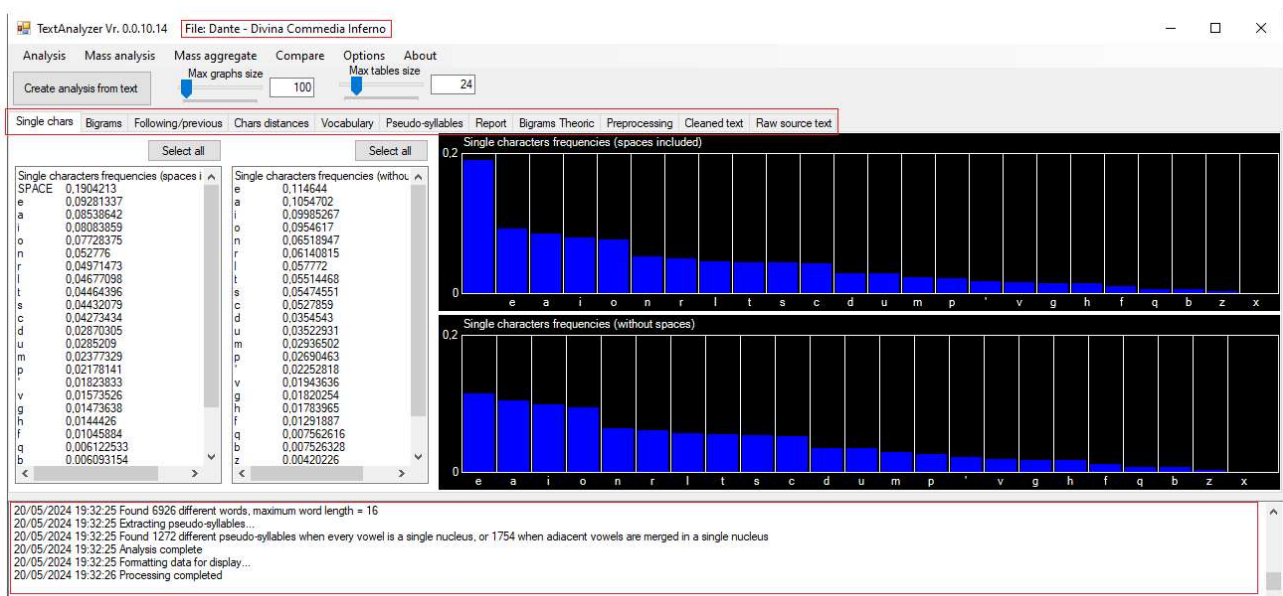
Salvataggio/caricamento di un’analisi

I files di analisi hanno estensione .txanalysis, sono dei normali files XML e possono essere aperti con un qualsiasi editor XML.

Quando si salva un’analisi viene prima aperta una finestra nella quale è possibile settare alcune informazioni opzionali (titolo del testo, autore, anno, lingua e annotazioni).

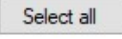
“Analysis file header infos” permette di visualizzare e modificare le informazioni opzionali.

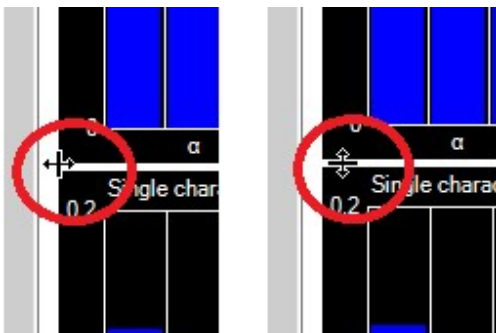
Pagina di analisi



Parti evidenziate dai riquadri rossi:

- Il nome del file analizzato compare nell'intestazione della finestra.
- Le tabs permettono di scegliere la visualizzazione delle varie statistiche ed informazioni.
- Sul fondo la 'status window' fornisce informazioni passo-passo sulle attività svolte da TextAnalyzer. In particolare, durante un'analisi vengono scritte varie informazioni che possono essere utili (per esempio: che opzioni sono state usate durante l'analisi ed il loro effetto, vedi "Pre-processing" per maggiori informazioni).

In generale, tutte le visualizzazioni sono divise in due parti: sulla sinistra vengono presentati dei valori numerici in formato testuale, sulla destra ci sono i grafici ricavati dagli stessi valori. I valori testuali sono concepiti per poter essere selezionati facilmente tramite i pulsanti  (o semplicemente con Ctrl-A) per poter poi essere copiati (Ctrl-C) ed incollati (Ctrl-V) nel programma che si preferisce (per esempio Excel) per ulteriori elaborazioni. Tutte le varie sottofinestre possono essere ridimensionate a piacere tramite gli "splitters" che si trovano fra una sezione e l'altra:



Bug conosciuto: se si è già cliccato all'interno di una finestra di dati il pulsante "Select All" non funziona più. Selezionare manualmente i dati all'interno della finestra, oppure ricaricare l'analisi per riabilitare i pulsanti.

Nota: i testi visualizzati nelle finestre appaiono spesso disallineati, ma questo accade perché sono concepiti per essere copiati in Excel (dove saranno perfettamente allineati).

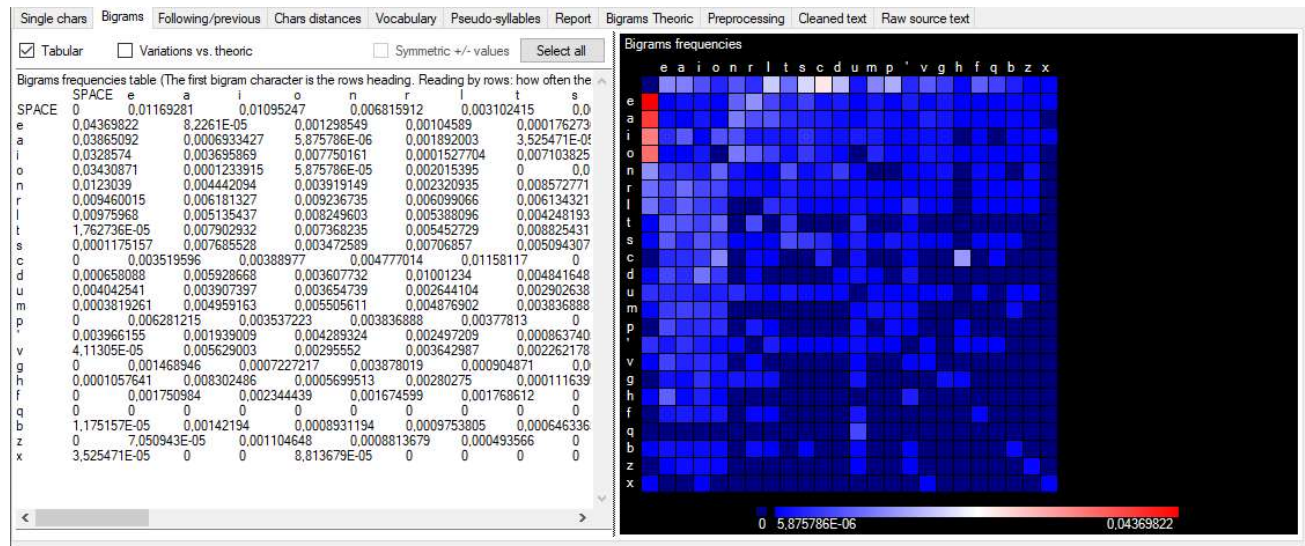
Tabs pagina analisi

Single chars

Frequenza dei singoli caratteri nel testo (spazio incluso o spazio escluso). Vedi il paragrafo precedente per la schermata.

Nota: dato che si tratta di frequenze la somma dei valori di tutti i caratteri dà 1.

Bigrams



Frequenze dei bigrammi nel testo (sempre inclusivi del carattere spazio, per esempio SPAZIO-a, SPAZIO-b etc.). Sia la tabella testuale che il grafico si leggono per righe: per esempio il bigramma “ae” si trova all’intersezione della riga “a” con la colonna “e”. Righe e colonne della tabella sono ordinate seguendo la frequenza dei caratteri singoli.

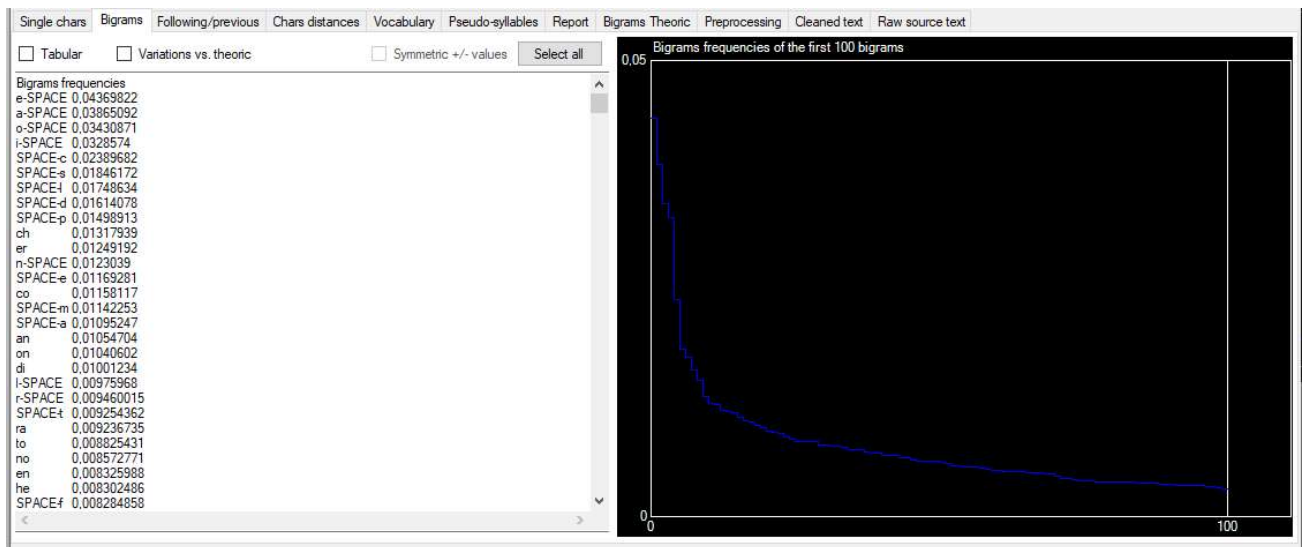
Il grafico bidimensionale usa una scala colori per rappresentare i valori, passando dal blu scuro al bianco e poi al rosso man mano che la frequenza aumenta. Ai bigrammi con frequenza zero (i bigrammi che non vengono mai usati nel testo) viene assegnato un colore blu particolarmente scuro in modo da poterli distinguere facilmente.

Nota. La scala dei colori viene calcolata automaticamente, assegnando il rosso al valore massimo contenuto nella tabella e il blu al valore minimo (entrambi questi valori possono essere letti guardando la scala) in modo che il grafico abbia la massima risoluzione possibile. Però questo non consente di comparare direttamente fra loro, guardando i colori, grafici derivati da testi diversi, perché in generale avranno valori minimi e massi diversi. Per farlo usare le funzioni di Comparazione, che usano una scala unica per tutti i grafici.

Nota: il carattere ‘spazio’ viene indicato con SPACE nella finestra coi dati testuali, ma non lo si vede (dato che è uno spazio) nelle intestazioni del grafico. Nell’esempio qua sopra la prima riga e la prima colonna rappresentano il carattere ‘spazio’ (questo accade quasi sempre, dato che ‘spazio’ è quasi sempre il carattere più frequente, tranne che in testi molto particolari)

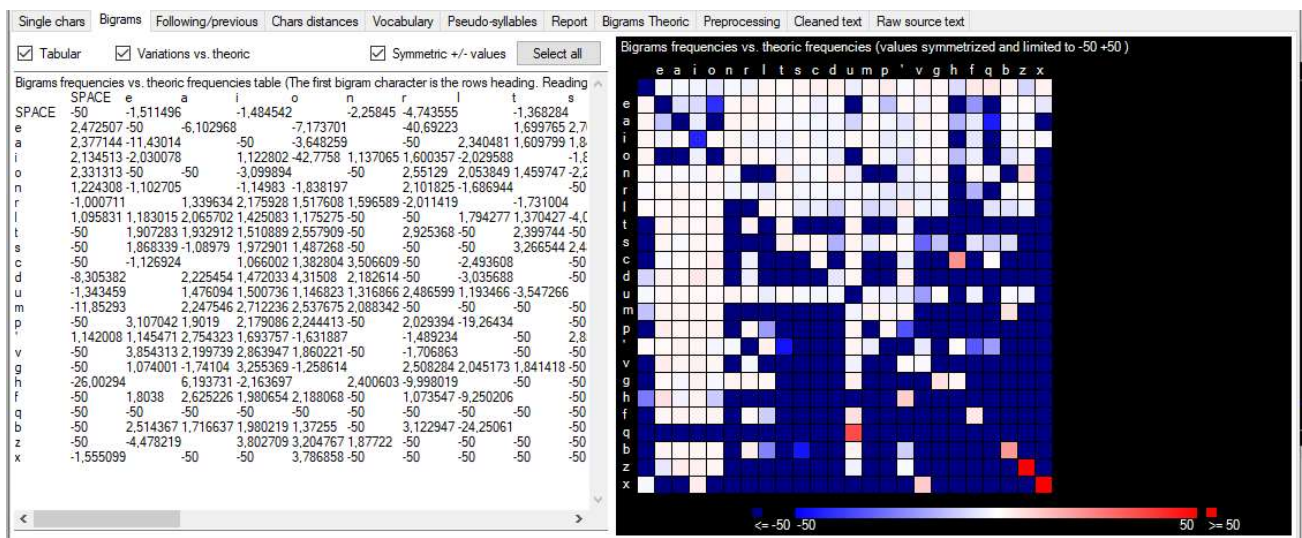
Nota: dato che si tratta di frequenze, la somma dei valori di tutti i bigrammi dà 1.

Togliere la spunta alla casella ☐ Tabular trasforma la visualizzazione da 2D ad 1D:



Adesso le frequenze dei bigrammi sono elencate in una lista ordinata dalla frequenza più alta a quella più bassa (cosa che può tornare utile in molti casi), ed anche il grafico diventa monodimensionale.

Spuntare ☒ **Variations vs. theoric** modifica profondamente l'aspetto dei dati:



Con questa visualizzazione le frequenze dei bigrammi vengono divise per le loro 'frequenze teoriche'. Le frequenze teoriche (che possono essere viste nella tab "Bigrams Theoric") sono ricavate dalle frequenze dei singoli caratteri ipotizzando che ad un carattere ne segua ad un altro in modo completamente casuale. Dividendo le frequenze effettive per quelle teoriche si vede quanto un certo bigramma sia 'preferito' o 'sfavorito' all'interno della lingua. Nell'esempio qua sopra si vede che il bigramma "qu" è favorito, il che è una conseguenza del fatto che nella grammatica italiana una "q" è (quasi) sempre seguita da una "u". Anche "ch" è favorito e questo dipende dal fatto che le parole in cui "c" è seguita da "h" (per esempio "chi", "che", "perché") sono molto frequenti. Un bigramma fortemente favorito è "zz", che indica che in Italiano la "z" è molto più spesso raddoppiata che singola.

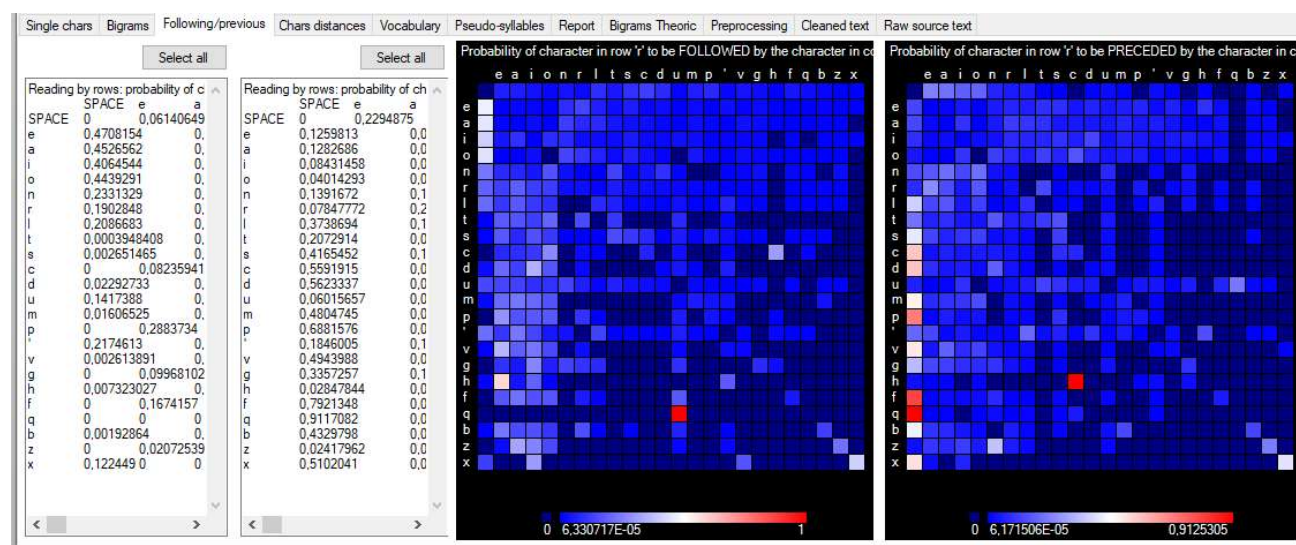
Nota: anche “xx” è estremamente favorito, ma questo è un dato anomalo, dovuto al fatto che “x” è un carattere molto raro e che compare quasi sempre seguito da un'altra “x”. Si tratta in effetti dei numeri dei canti della Divina Commedia, scritti come numeri romani. Vedere il paragrafo “Sliders per settare i limiti dei grafici” per eliminare i caratteri rari dalla visualizzazione.

Dividendo le frequenze effettive per quelle teoriche si ottengono dei numeri maggiori di 1 (che possono anche essere piuttosto grandi) per i bigrammi che vengono ‘amplificati’, mentre si ottengono valori minori di 1 per i bigrammi ‘soppressi’. La scala del grafico andrebbe da circa zero fino ad un valore piuttosto grande, quindi visualizzando i valori come tali i bigrammi ‘soppressi’ avrebbero tutti un colore blu scuro e l’unica cosa che spiccherebbe sarebbero i valori particolarmente amplificati. Per questo motivo, dei valori minori di 1 viene preso il reciproco, in modo da riportarli in un range simile a quello dei bigrammi amplificati, e gli viene cambiato il segno in modo da poterli distinguere. Inoltre, i valori minimo e massimo vengono limitati a ± 50 .

Nota: una casellina per poter settare i limiti minimo e massimo potrebbe essere utile, ma credo richieda più lavoro di quanto si guadagni in utilità.

Nota: i valori grezzi che si ottengono dalla divisione possono essere visualizzati, se lo si desidera, togliendo la spunta a ☒ Symmetric +/- values

Following/Previous

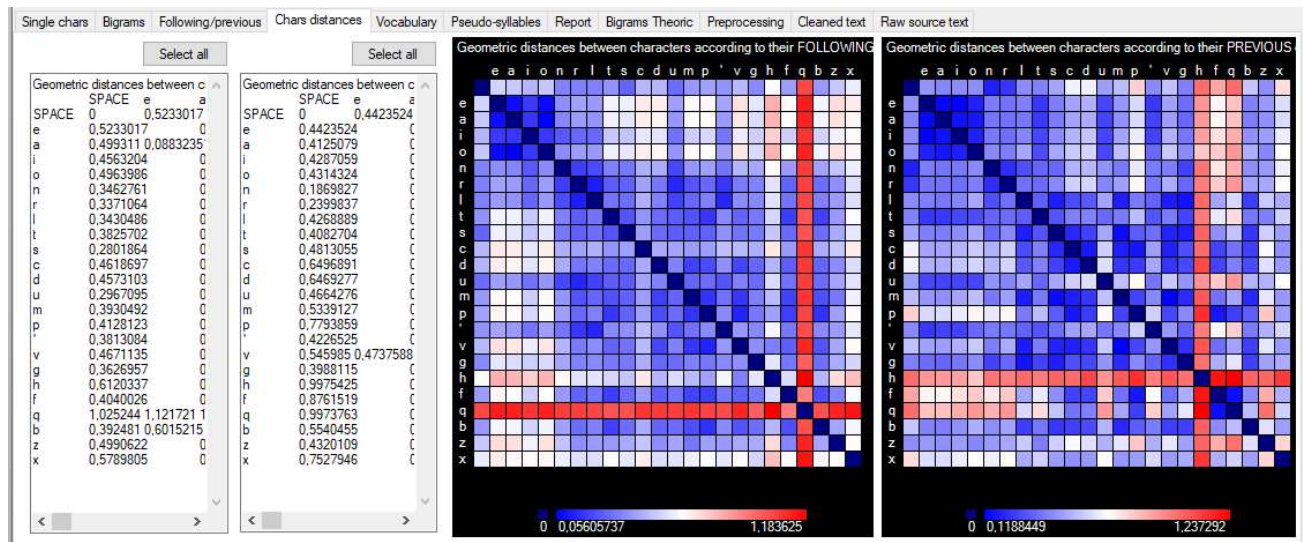


Partendo dalle frequenze dei bigrammi, per ogni carattere vengono calcolate la probabilità che sia seguito da un certo altro carattere (per esempio: probabilità che una “e” sia seguita da una “a”) e la probabilità che sia preceduto da un certo altro carattere (per esempio: probabilità che una “e” sia preceduta da uno “spazio”).

Questa visualizzazione è utile per analisi ortografiche: per esempio dal grafico “FOLLOWED si vede facilmente come in Italiano (Dante-Inferno) la “q” sia sempre seguita da “u” mentre, dal grafico “PREVIOUS” si vede che la “q” è molto spesso ad inizio parola. Inoltre la “h” è molto spesso preceduta da “c” (si tratta delle parole “chi”, “che” “perché” etc. già notate prima). La “f” e, in misura minore, “c” e “d” tendono ad essere ad inizio parola.

Nota: in queste tabelle la somma di ogni riga dà 1.

Chars distances

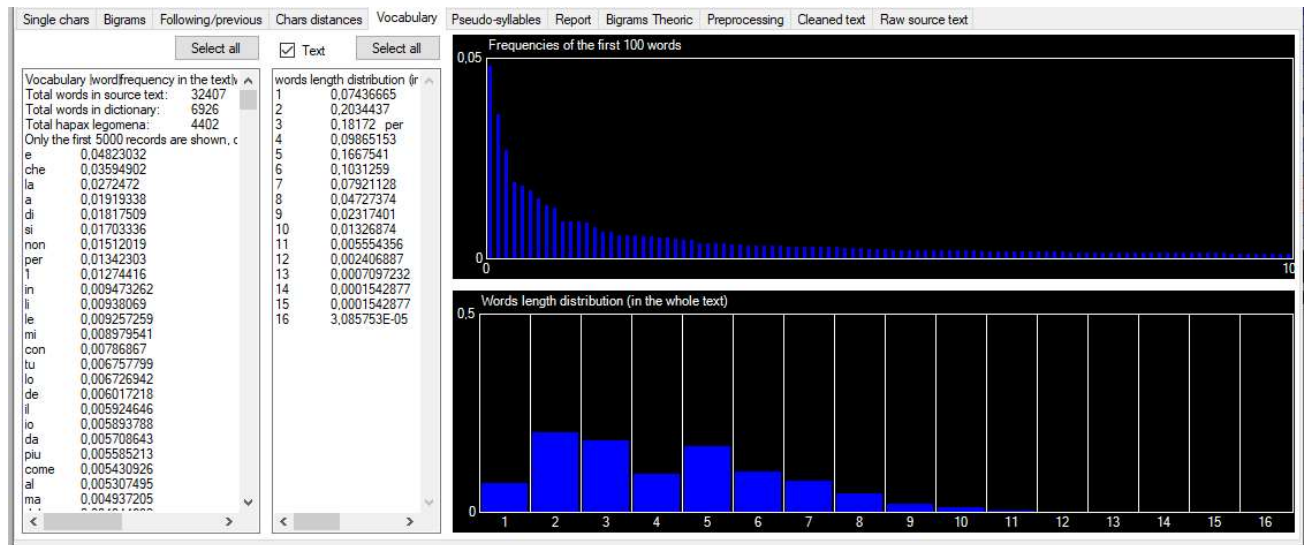


Partendo dalle probabilità del paragrafo “Following/Previous”, viene calcolato quanto ogni carattere è ‘distante’ dagli altri. Per ‘distanza’ si intende la distanza geometrica (Euclidea) fra due righe delle tabelle: per esempio la distanza fra “e” ed “a” sulla base del carattere seguente è calcolata come la radice quadrata della somma dei quadrati delle differenze fra ogni cella della riga “e” e la corrispondente cella della riga “a” nella tabella “Probabilità che un carattere sia seguito da un certo altro carattere”.

In pratica queste distanze indicano quanto un carattere possa ‘sostituirne’ un altro (statisticamente) in una parola. Sia grafico “FOLLOWING” che in quello “PREVIOUS” qua sopra vediamo, per esempio, che “e” è molto simile ad “a”: questo vuol dire che se esiste la parola “mare” è probabile che esisteranno anche le parole “mara”, “mera” e “mere”. La lettera “q”, invece è diversa da tutte le altre per quanto riguarda il carattere successivo (e, in misura minore, anche per il carattere precedente), mentre la “h” è molto diversa da tutte le altre per quanto riguarda il carattere precedente (infatti in Italiano è preceduta solo da “c”, “g” o “spazio”, salvo i rari casi delle interiezioni “eh”, “oh” eccetera).

Nota: dato che si tratta di distanze, queste due tabelle sono simmetriche e sulla diagonale la distanza è sempre zero (è la distanza di un carattere da sé stesso). Il valore massimo che si può trovare in tabella (due caratteri il più distanti possibili l’uno dall’altro, tecnicamente: due vettori ortogonali) è $\sqrt{2} \approx 1.414$.

Vocabulary



Sulla sinistra troviamo il vocabolario estratto dal testo, ordinato secondo la frequenza con cui ogni parola compare.

Nota: per poter estrarre il vocabolario TextAnalyzer deve supporre che il carattere 'spazio' rappresenti effettivamente il separatore fra le parole. Non è detto che questo sia sempre vero, credo sia comunque probabile che il carattere 'spazio' sia sempre quello più frequente in tutte le lingue (è così per tutte quelle che ho esaminato ma non posso escludere che esistano lingue in cui questo non è vero).

Nota: di default vengono visualizzate solo le prime 5000 parole del vocabolario, per risparmiare tempo di esecuzione ed avere una risposta più pronta ai comandi. Vedere "Opzioni di visualizzazione Analisi" per abilitare la visualizzazione di tutte le parole, e le precauzioni per l'uso.

Nota: TextAnalyzer non 'conosce' alcunchè di nessuna lingua (è stato realizzato per essere il meno specializzato possibile e per poter essere applicabile a qualsiasi lingua), quindi considera ogni sequenza di caratteri distinta come una 'parola' diversa. Per esempio, tutte le voci del verbo "andare" sono considerate parole distinte. Il carattere "apostrofo" è particolarmente problematico ed esistono alcune opzioni per poterlo trattare (vedi "Pre-processing base e Opzioni di pre-processing" per una discussione completa): qua viene considerato come un carattere normale a tutti gli effetti, quindi nel vocabolario compaiono 'parole' contenenti apostrofi (per esempio " 'l " qua sopra), e 'parole' come "l'amore" o "un'amore". La lunghezza del vocabolario, quindi, può essere molte volte maggiore di quella di un vocabolario standard.

Viene anche calcolata la distribuzione della lunghezza delle parole considerando l'intero testo, oppure, togliendo la spunta alla casellina ☒ Text, la distribuzione della lunghezza delle parole nel vocabolario (che è in generale molto diversa):

Nota: esistono comunque degli algoritmi che sono in grado di identificare abbastanza bene quali simboli rappresentino una vocale e quali una consonante. Quindi, in teoria, il problema dell'elenco delle vocali potrebbe essere superato.

Definito quali caratteri siano vocali e quali consonanti bisogna stabilire quale sia il “nucleo” di una sillaba (le sue vocali) e qua sorge subito un'ambiguità, perchè da lingua a lingua (e spesso anche da parola a parola in una stessa lingua) una singola vocale può essere un nucleo separato, oppure può essere parte di un dittongo vocalico che va inserito così com'è nel nucleo. In generale non è possibile risolvere queste ambiguità (e altre nel seguito) senza conoscere approfonditamente una lingua, cosa che TextAnalyzer non può (e non vuole) fare. Quindi non c'è da aspettarsi che la sillabazione sia sempre ‘esatta’, per quanto nella pratica gran parte delle parole vengano sillabate correttamente.

Nota: farei anche notare che le sillabe sono un concetto fonetico più che ortografico, e che le idiosincrasie ortografiche di ogni lingua nel rappresentare i fonemi rendono il problema spesso intrattabile. Si potrebbero ottenere risultati praticamente perfetti in lingue che hanno un'ortografia molto coerente con la fonetica (credo le lingue slave, ma forse anche il tedesco), ma già con l'Italiano (nonostante in Italia si creda che si “legge come si scrive”) esistono una quantità di casi impossibili da risolvere se non si conosce a fondo la lingua (per esempio la “i” di “piacere” *non* rappresenta una vocale). In una lingua come l'Inglese i casi impossibili si moltiplicano, per esempio la “y” è di solito una consonante, ma in “happy” è una vocale.

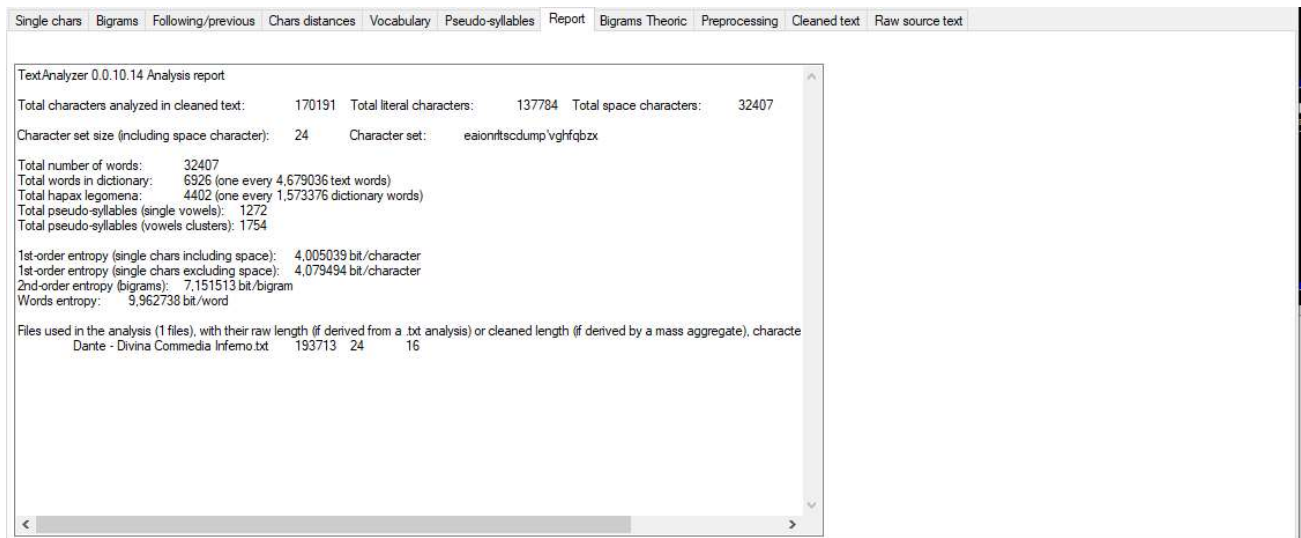
I nuclei vengono determinati in due modi diversi: ogni vocale forma un nucleo a sé (“mai” viene sillabato “ma-i”) oppure tutte le vocali adiacenti vengono riunite in un nucleo (“mai” viene sillabato “mai”).

Nota: la seconda possibilità è spesso quella che dà i risultati migliori.

Trovato il nucleo, per completare le sillabe con le consonanti bisognerebbe conoscere come minimo la loro “scala di sonorità” (non spiego qua che cos'è). Aggiungere anche questa sposterebbe molto TextAnalyzer dall'essere “language-independent”, per cui uso un semplice algoritmo che spezza le consonanti adiacenti in due gruppi e ne assegna metà alla sillaba precedente e metà alla successiva, dando priorità alla sillaba successiva. Questo sillaba correttamente parole come “rara” (ra-ra) o “strambo” (“stram-bo”) o, in Inglese, “sensation” (sen-sa-tion). L'algoritmo fallisce nei casi in cui la scala di sonorità sarebbe stata importante, per esempio “madre” viene sillabata, erroneamente, mad-re. Fallisce anche in casi in cui ci sono delle idiosincrasie ortografiche, per esempio “lasciare” viene sillabato las-cia-re dato che textAnalyzer non ha modo di sapere che “sci”, in realtà, rappresenta un unico suono.

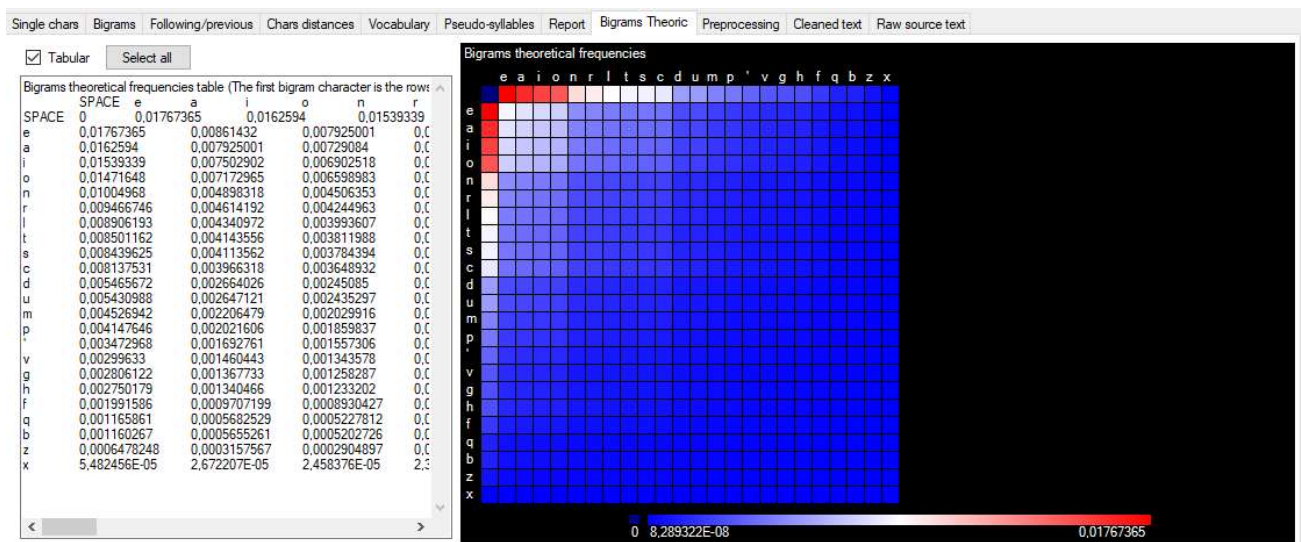
Nota: analogamente al Vocabolario, di default vengono visualizzate solo le prime 5000 sillabe, per risparmiare tempo di esecuzione ed avere una risposta più pronta ai comandi. Vedere “Opzioni di visualizzazione Analisi” per abilitare la visualizzazione di tutte le sillabe, e le precauzioni per l'uso. Attenzione in particolare alle lingue che non possono essere sillabate perché manca la definizione delle loro vocali, dato che in questo caso il numero delle sillabe diventa uguale a quello della parole e può essere molto grande.

Report



Fornisce una serie di informazioni aggiuntive sul testo.

Bigrams theoretic



Frequenze teoriche dei bigrammi, ricavate dalla distribuzione dei caratteri singoli supponendo che ad un carattere ne segua un altro in modo completamente casuale. Vengono utilizzate nel calcolo delle “variations vs. theoretic”, vedi paragrafo “Bigrams”.

Preprocessing, Cleaned text e Raw text

La tab Preprocessing visualizza eventuali commenti o comandi inseriti dall'utente nel testo, vedi il paragrafo "Commenti e comandi Regex".

Cleaned text e Raw text visualizzano il testo come è stato caricato dal file .txt (Raw text) e quello ottenuto dopo il preprocessing (Cleaned text, vedi "Errore. L'origine riferimento non è stata trovata.").

Nota: per risparmiare tempo di esecuzione i testi vengono visualizzati solo se contengono meno di 300K caratteri (viene però visualizzato un messaggio esplicativo). Vedere "Opzioni di visualizzazione Analisi" per abilitare la visualizzazione dell'intero testo.

Nota: di default il testo *non* viene salvato nel file di analisi (.txanalysis). Quindi caricando un file di analisi non viene visualizzato alcun testo (viene visualizzato un messaggio esplicativo).

Sliders per settare i limiti dei grafici

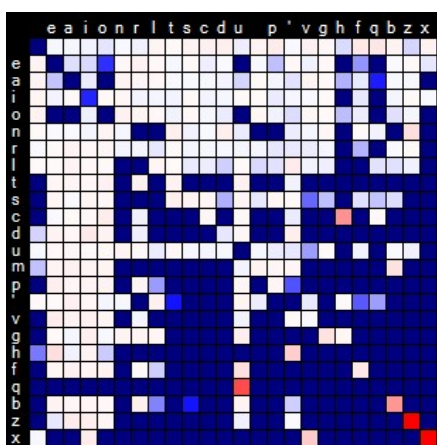
È possibile definire cosa visualizzare nei grafici tramite gli sliders "Max graphs size" e "Max tables size" (o le caselline numeriche associate):



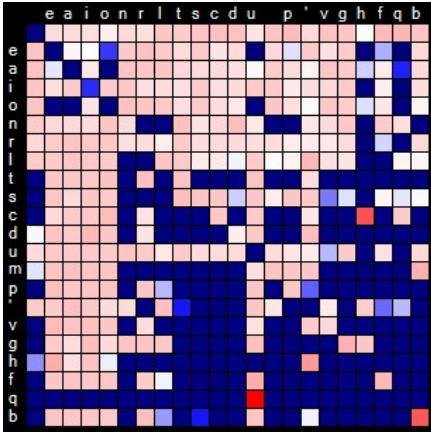
"Max graphs size" controlla i grafici monodimensionali (gli istogrammi) e consente di visualizzare nel grafico da un minimo di 10 ad un massimo di 2000 elementi. Il default è 100.

"Max tables size" controlla la dimensione dei grafici bidimensionali. Defaulta al numero di caratteri presenti nel testo ed è particolarmente utile per rimuovere dai grafici le lettere rare.

Notare che per entrambi i tipi di grafici la scala viene ricalcolata sul numero di elementi effettivamente visualizzati: questo consente di vedere più nel dettaglio i dati rimanenti. Per esempio, le frequenze dei bigrammi "Variation vs. theoric" della Divina Commedia – Inferno sono:

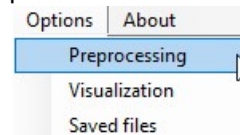


La scala colori è condizionata dai due quadratini rossi dei bigrammi “zz” e “xx”, e questo comprime la scala di tutti gli altri bigrammi. Usando lo slider “Max tables sizes” per rimuovere “x” e “z” si ottiene in grafico i cui le variazioni dei caratteri rimanenti sono più in evidenza:

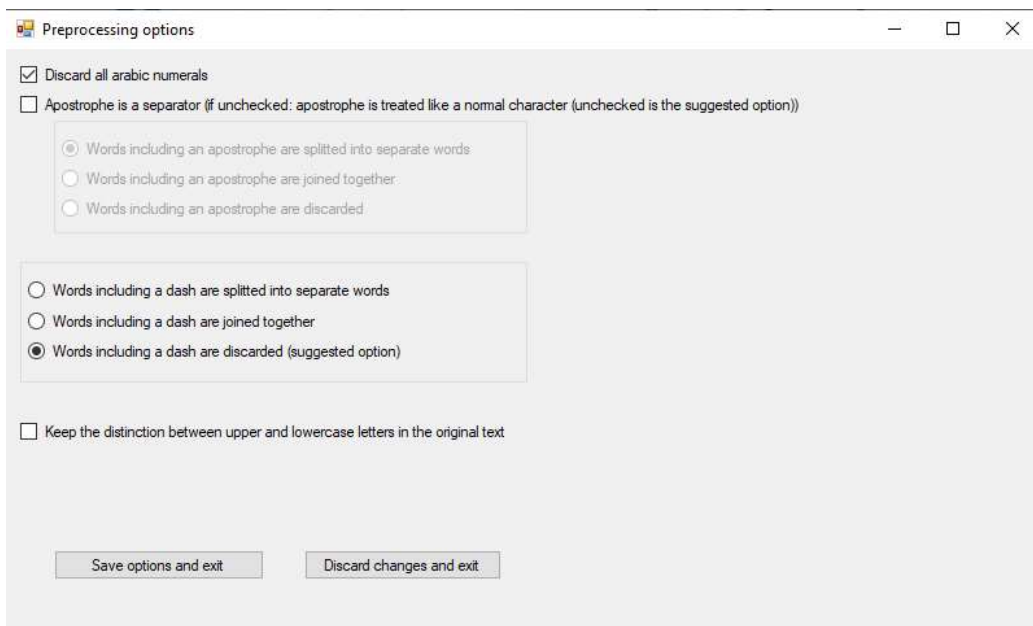


Pre-processing

Prima dell’analisi un testo viene ‘ripulito’ il più possibile da vari elementi spuri. Questa operazione si chiama “pre-processing” e converte il testo così come caricato (il “raw text”) in un testo ‘pulito’ (il “cleaned text”) sul quale viene eseguita l’analisi. Il pre-processing ha lo scopo di evitare tediose ‘ripuliture’ manuali dei testi e di

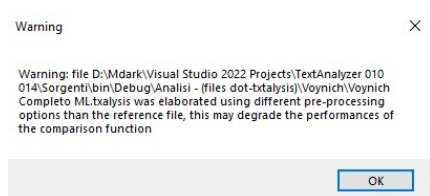


standardizzare il trattamento di alcuni aspetti particolari. Tramite il menù modificare il modo con cui il pre-processing tratta certi caratteri:



Consiglierei di usare sempre le stesse opzioni per tutti i testi, dato che questo rende poi più affidabile la comparazione di un testo con un altro. Non è però un obbligo (vedi anche nota al paragrafo “Gestione dell’apostrofo”).

Nota: le funzioni di comparazione (vedi in seguito) danno un avviso quando si cerca di comparare testi elaborati con opzioni differenti:



Pre-processing base e Opzioni di pre-processing

L’opzione “Discard all arabic numerals” elimina tutti i numeri arabi dal testo (vengono sostituiti con ‘spazi’). È attiva di default, si potrebbe volerla togliere se in un testo i numeri arabi rappresentassero dei veri e propri caratteri.

Di default TextAnalyzer trasforma tutti i caratteri maiuscoli in minuscoli. In casi particolari si potrebbe voler mantenere la distinzione, cosa che si può fare settando l’opzione “Keep the distinction between upper and lowercase letters in the original text”.

Gestione del trattino (‘dash’)

Il trattino ‘-’ ha un uso incostante fra un testo e un altro. Sono previste tre opzioni diverse ma, in generale, la migliore è quella che fa scartare le parole contenenti un trattino (“Words including a dash are discarded”).

I trattini vengono normalmente usati per:

1. Indicare un ‘andare a capo’. Spesso, nei testi elettronici ricavati da testi originariamente non elettronici c’è stata una rimpaginazione e il trattino ha perso la sua funzione: è presente ma la parola non è più spezzata dal carattere “a capo”, per esempio si può trovare una riga che contiene “vi-sione”, dove nel testo originale c’era un cambio di riga fra “vi” e “sione” (caso 1a). Altre volte, però, il cambio di riga è stato mantenuto (caso 1b, il caso peggiore).
2. Enfasi: “e-vi-den-te-men-te!”
3. Enfasi, ma in un modo diverso: “quel buono-a-nulla!”

Una delle opzioni disponibili è “Words including a dash are joined together”. Funziona benissimo nel caso 1a), ma non nel caso 1b) dato che il carattere “-” spezza comunque la parola in due parti (verrebbero accettate sia la parola “vi” che la parola “sione”. Funziona benissimo anche nel caso 2), ma il caso 3) rappresenta un problema perché si rischia di generare parole spurie lunghissime nel caso l’autore si sia divertito coi trattini (in un testo ha generato una parola di 54 lettere, che in originale era scarpe-antisdrucchiolo-del-Nord-Dakota- etc. etc.)

Un’altra opzione è “Words including a dash are separated”, che funziona benissimo nel caso 3) ma genera parole spurie (e corte) nei casi 1) e 2).

L’opzione suggerita è “Words including a dash are discarded”: elimina delle parole dal testo ma evita di generare parole spurie e di gonfiare inutilmente il vocabolario. L’unico caso in cui ha dei problemi è il caso 1b):

nell'esempio "vi-" a capo "sione" la parola "vi" verrebbe eliminata, ma verrebbe accettata la parola spuria "sione".

Nota: i testi di tipo 1b) sono abbastanza rari, suggerisco quindi, semplicemente, di evitare di includerli in un corpus. A parte il problema delle parole spurie (che è spiacevole), comunque, i trattini sono in generale rari di per sé e quindi influiscono poco sulle statistiche.

Gestione dell'apostrofo

Gli apostrofi sono il carattere più problematico per una gestione "language-independent":

1. Esistono lingue dove l'apostrofo è un carattere vero e proprio, che rappresenta un suono effettivo (spesso lo "stop glottale", per esempio in Hawaiano). In queste lingue è normale che una parola contenga un apostrofo.
2. In una lingua come l'Inglese l'apostrofo è usato per indicare un'elisione. Alcune volte (per esempio con "don't", "I'm" etc.) sarebbe naturale accettare le parole così come sono e inserire "don't", "I'm" etc. nel vocabolario. Ma nel caso del genitivo sassone ("John's", "Mary's") verrebbero generate delle parole spurie.
3. Anche in Italiano l'apostrofo è usato per indicare un'elisione, ma in modo molto più massiccio, cosa che moltiplica le dimensioni del problema rispetto all'Inglese.

L'opzione suggerita per l'apostrofo è di considerarlo sempre un carattere vero e proprio. Non è affatto esente da difetti, ma consente quasi sempre risultati migliori che settare "Apostrophe is a separator" (che a sua volta consente poi tre scelte diverse, simili a quelle per il trattino).

1. Non causa alcun problema con lingue come l'Hawaiano (anzi, in questo caso è l'unica opzione).
2. In Inglese accetta le parole "don't", "I'm" etc., che mi sembra una cosa corretta dato che sono molto caratteristiche dell'Inglese. Come svantaggio, inserisce nel vocabolario parole spurie come "John's" e "Mary's" (l'uso del genitivo sassone è comunque poco frequente).
3. In Italiano al vocabolario vengono aggiunte molte parole spurie contenenti apostrofi, per esempio "l'amore", "un'amore", "all'amore", "dell'amore".

In Italiano il problema delle parole spurie (e quindi delle dimensioni raggiunte dal vocabolario) è piuttosto grave: qua potrebbe essere invece indicata l'opzione "Apostrophe is a separator -> Words including an apostrophe are discarded", che non introduce parole spurie ma, data la frequenza con cui l'apostrofo è usato in Italiano, al costo di eliminare una quantità abbastanza sensibile di parole incluse 'parole' comuni come "c'è" o "l'ho".

In conclusione: non esiste un modo per trattare gli apostrofi che sia "language-independent" ed esente da difetti. Usando l'opzione suggerita il difetto principale è l'aggiunta al vocabolario, in una lingua come l'Italiano, di una quantità di parole spurie.

Nota: TextAnalyzer è stato concepito per poter essere completamente agnostico su quale sia la lingua di un testo, per questo motivo ho consigliato precedentemente di usare sempre le stesse opzioni per tutte le analisi. Avendo di fronte una lingua sconosciuta non si ha neanche modo di sapere cosa un apostrofo, per esempio, possa rappresentare, e l'uniformità nell'uso delle opzioni dovrebbe poi rendere più affidabili le comparazioni fra testi diversi.

Questo però non è assolutamente un obbligo e, conoscendo qualcosa sulla lingua del testo che si vuole analizzare, si possono scegliere le opzioni che si preferiscono (ma ricordarsi di pensare bene a quali sono le conseguenze).

Nota: il carattere 'apostrofo' è un carattere ben specifico (codice U+0027) e, sperabilmente, nei testi è stato usato quello. Capitano però casi in cui al posto dell'apostrofo 'standard' è stato usato un carattere che gli assomiglia graficamente ma che non è un apostrofo, per esempio ' (codice U+2019) o ' (codice U+2018). TextAnalyzer sostituisce automaticamente U+2019 (il più comune). Se si trovasse U+2018 sostituirlo nel testo con quello standard (può farlo anche TextAnalyzer stesso: vedi paragrafo "Commenti e comandi Regex").

Commenti e comandi Regex

All'interno del file di testo è possibile aggiungere delle linee di commento, che non verranno considerate nell'analisi. Una linea è considerata un commento se inizia per %%:

`%% Questo è un commento`

I commenti possono essere usati per qualsiasi informazione si voglia aggiungere al testo, ma con un tipo particolare di commento si può fare in modo che TextAnalyzer esegua delle operazioni di search-and-replace sul testo prima di analizzarlo. Questi commenti particolari sono chiamati "Comandi Regex" e hanno questa forma:

`%%REGEX;stringa search;stringa replace;commento`

Il formato è rigido: al %% deve seguire immediatamente la parola REGEX (in maiuscolo) seguita da un punto e virgola, a cui segue la stringa di ricerca terminata da un altro punto e virgola. Si prosegue con la stringa di replace, sempre terminata da punto e virgola, e alla fine si può scrivere un commento libero. Per esempio, questo comando trasforma il carattere 'ᾱ' in 'α', eliminando il diacritico:

`%%REGEX;ᾱ;α;Rimozione diacritici greci`

Il search-and-replace utilizza una funzionalità software standard chiamata Regular Expressions ("Regex") che è estremamente potente e può eseguire ogni genere di operazioni su un file di testo. Essendo una funzione molto potente, però, è anche pericolosa da usare se non si sa quello che si sta facendo. Finché ci si limita a sostituire dei caratteri con altri caratteri non si dovrebbero ottenere effetti imprevisti, ma evitate di inserire caratteri di interpunzione, parentesi, '+', '\', e altre stranezze nella stringa 'search' se non sapete che cosa significano per Regex perché i risultati potrebbero essere ben diversi da quello che ci si attende, per esempio:

`%%REGEX;. ; Sostituisce il punto con uno spazio !?!`

Non sostituisce i punti con spazi, ma rimpiazza tutti i caratteri del testo con spazi!

Nota: la sintassi completa dei comandi Regex è tutt'altro che semplice, se siete interessati la si trova in molti siti su Internet, per esempio qua: <https://learn.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference>. Un sito molto utile per sperimentare con Regex è <https://regex101.com/>. Questo sito è molto utile anche per vedere che codifica abbia un certo carattere (per esempio nel caso si abbiano problemi con caratteri 'strani' ma difficili da distinguere graficamente dai caratteri 'normali').

Nota: con Regex è possibile lanciare comandi che eseguono elaborazioni complesse e che possono richiedere molto tempo di esecuzione o molta memoria (non è questo il caso delle sostituzioni semplici, a meno di files di testo enormi, contenenti molte decine di milioni di caratteri). In questo caso Regex può fallire l'elaborazione per mancanza di memoria o perché l'elaborazione richiede troppo tempo (timeout). TextAnalyzer avvertirà nella status window che il comando Regex non è andato a buon fine.

I comandi REGEX possono essere usati per una quantità di cose utili, per esempio:

- Rimuovere i diacritici dai testi in Greco (tenete presente che ce ne sono centinaia... ma di comandi Regex se ne possono aggiungere quanti se ne vuole, uno per ogni diacritico che si trova).
- Trasformare un testo in Tedesco con diacritici in un testo senza diacritici, o viceversa. Per esempio questo comando rimpiazza "ü" con "ue":

`%%REGEX;ü;ue;Rimozione diacritici Tedesco`

- Trasformare un apostrofo non-standard in un apostrofo standard:

`%c%REGEX;' ; ' ;` Sostituisce apostrofo non-standard U+201B con standard U+0027

- Qualsiasi altra cosa possa tornare utile alla vostra analisi.

Nota: i comandi Regex vengono lanciati dopo che sono state gestite le altre opzioni di pre-processing. Questo vuol dire, per esempio, che è possibile eliminare tutte le maiuscole da un testo e poi introdurre delle maiuscole tramite i comandi Regex, cosa che potrebbe essere utile in certi casi (per esempio si potrebbe voler sostituire il “ch” in Tedesco con un carattere speciale, dato che rappresenta sempre lo stesso suono: si potrebbe sostituirlo con “X” maiuscola in modo da distinguerlo facilmente).

Nota: una possibilità futura è aggiungere un altro tipo di commento che consenta di settare le opzioni di pre-processing direttamente all'interno del file di testo, senza dover modificare le opzioni globali.

Altre operazioni eseguite durante il pre-processing

- Vengono eliminate le abbreviazioni composte da una lettera seguita da un punto. Bug conosciuto: la gestione non è perfetta, abbreviazioni come N. o a.C. vengono eliminate completamente, ma per esempio q.o.d. o S.P.Q.R. non vengono eliminate completamente e restano delle lettere spurie (la ‘o’ di q.o.d e la ‘p’ di S.P.Q.R.).
- L’apostrofo non-standard U+2019 viene sostituito da quello standard U+0027
- Vengono eliminati (sostituendoli con ‘spazio’) alcuni caratteri particolari: underscore ‘_’ e grado ‘°’.
- Vengono eliminati tutti i segni di interpunzione, parentesi, rinvii a capo etc. (tutto quello che è un “non-word character” secondo la definizione di Regex), sostituendoli con spazi.
- Tutti gli spazi adiacenti vengono compattati in un unico spazio.

Messaggi di pre-processing nella status window

La status window fornisce informazioni sulle opzioni usate in un’analisi e sugli eventi durante il pre-processing

```
22/05/2024 09:53:47 Pre-processing loaded file...
22/05/2024 09:53:47 Using options:
22/05/2024 09:53:47     Discard all arabic numerals: True
22/05/2024 09:53:47     Apostrophe is a separator: False
22/05/2024 09:53:47     Words including a dash are discarded
22/05/2024 09:53:47     Keep the distinction between upper and lowercase: False
22/05/2024 09:53:47 Removed 0 characters representing arabic numerals
22/05/2024 09:53:47 Discarded 359 putative abbreviations (single characters followed by a dot)
22/05/2024 09:53:47 Removing words containing dashes and/or apostrophes, depending on the pre-processing options....
22/05/2024 09:53:47 Discarded 10 words containing a dash '-'
22/05/2024 09:53:47 Removing punctuation characters and some special character such as '°' (degree symbol), may take a while....
22/05/2024 09:53:47 Preprocessing complete
```

Codifiche dei files di testo e preparazione dei files di testo

I testi possono utilizzare ‘codifiche’ diverse per rappresentare i caratteri. La codifica utilizzata da Windows al giorno d’oggi è detta “UTF-8” ed è quella base con cui lavora TextAnalyzer. Può rappresentare praticamente tutte le lingue al mondo (eccettuato Cinese e Coreano, che usano una codifica specializzata che non è supportata da TextAnalyzer).

Usando Window Notepad è possibile vedere (e modificare) la codifica di un file usando la funzione “Salva con nome”:



Se si hanno dei files di testo in formato diverso da UTF-8 (può capitare con files poco recenti, o copiando pagine da Internet scritte con vecchie codifiche) è opportuno “salvarlo con nome” e modificare la codifica ad UTF-8, TextAnalyzer cercherà comunque di aprire anche i files UTF-7. L’uso di questa modalità viene segnalato nella status window:

22/05/2024 10:04:01 Loading file D:\Mdark\Visual Studio 2022 Projects\TextAnalyzer 010 014\Sorgenti\bin\Debug\Testi\Italiano\Boccaccio - Trattatello In Laude Di Dante.txt... probably UTF-7 encoded, trying opening it as such...

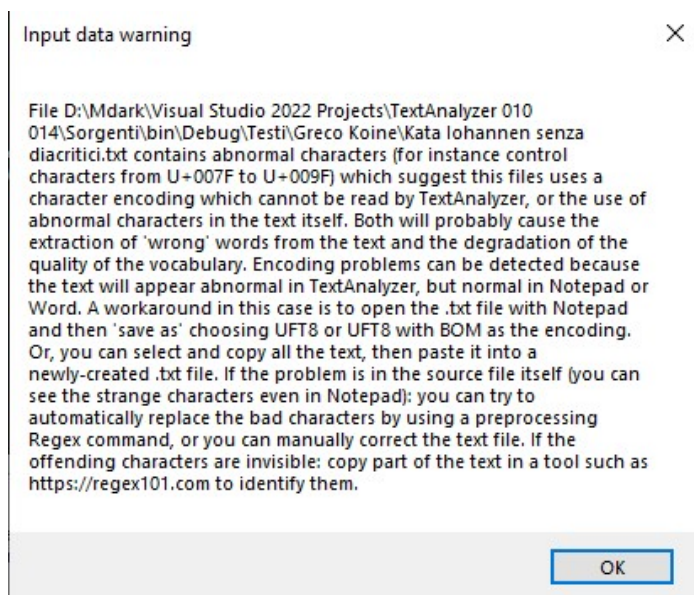
Se si hanno dei testi in formato .pdf, .doc o .rtf (anche se contenenti immagini o altri elementi non testuali) possono essere facilmente convertiti in testo aprendo il file col suo programma (per esempio Acrobat Reader per i .pdf, Word per i .doc e .rtf), usando Ctrl-A per selezionare tutto e poi Ctrl-C per copiare l’intera selezione. A questo punto si può aprire Windows Notepad e incollare tutto il testo con Ctrl-V.

Nota: procedendo in questo modo il file di testo viene salvato automaticamente come UTF-8

Risoluzione di problemi coi files di testo

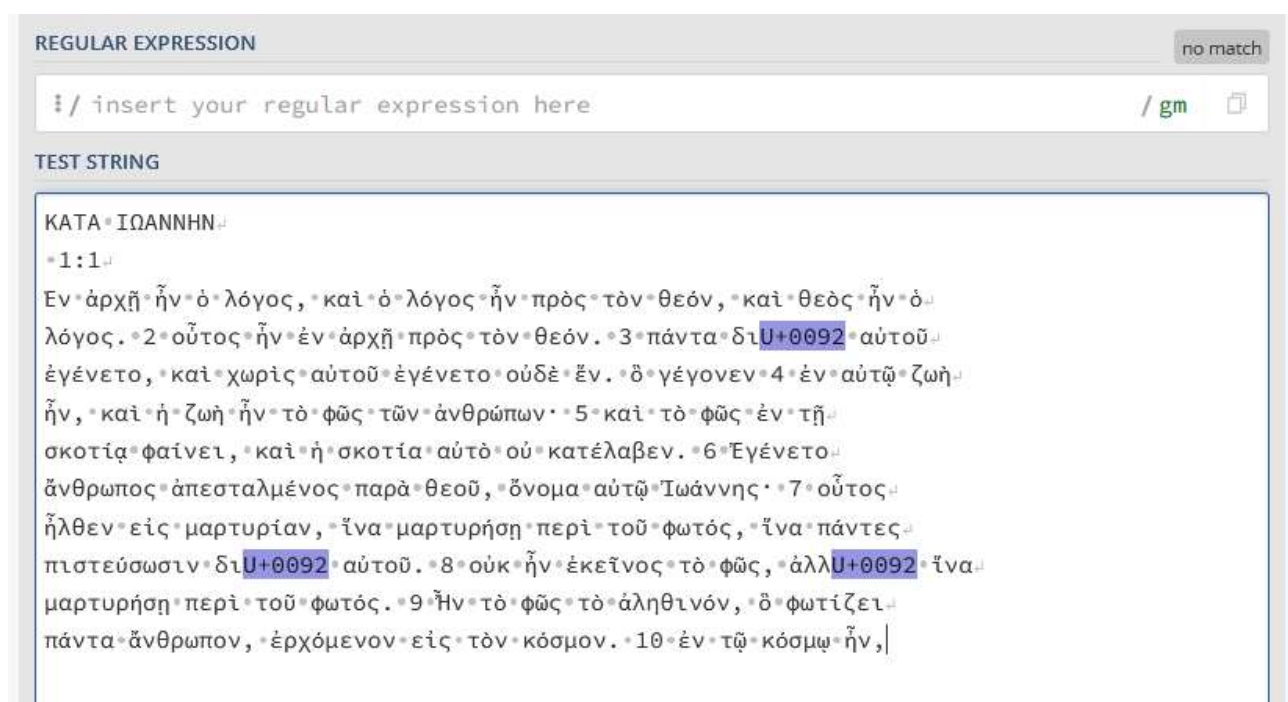
Un problema abbastanza comune è l’uso nei testi di caratteri ‘strani’. Può essere:

1. Un problema dovuto alla codifica originale del testo (probabilmente messo in forma elettronica molto tempo fa) che non è più compatibile con le codifiche moderne. Questo lo si riconosce facilmente perché si vedono caratteri ‘strani’ anche aprendo il file con Notepad (o Acrobat Reader o Word), per esempio i caratteri accentati potrebbero essere tutti sostituiti da caratteri ‘strani’. In questo caso c’è poco da fare, se non risistemare il testo manualmente (ma è raro che ne possa valere la pena). Un problema analogo lo si trova, a volte, con testi che indicano particolarità come il grassetto o il corsivo racchiudendo le parole fra simboli come In questo caso TextAnalyzer accetterebbe la “b” come una parola (dopo aver eliminato i segni di interpunzione), ma il problema è facile da sistemare rimpiazzando manualmente (o con comandi Regex) etc. con una stringa nulla (o con uno spazio).
2. Un problema dovuto a idiosincrasie (o magari a errori) dell’autore o dell’editore. Qua si trova di tutto, inclusi casi rognosi in cui i caratteri ‘strani’ non si vedono affatto (perché si tratta di caratteri di controllo, o di vere e proprie stranezze come il carattere “zero width space”), oppure sono graficamente indistinguibili dai caratteri normali. In particolare, se TextAnalyzer visualizza il messaggio seguente il testo contiene caratteri invisibili:



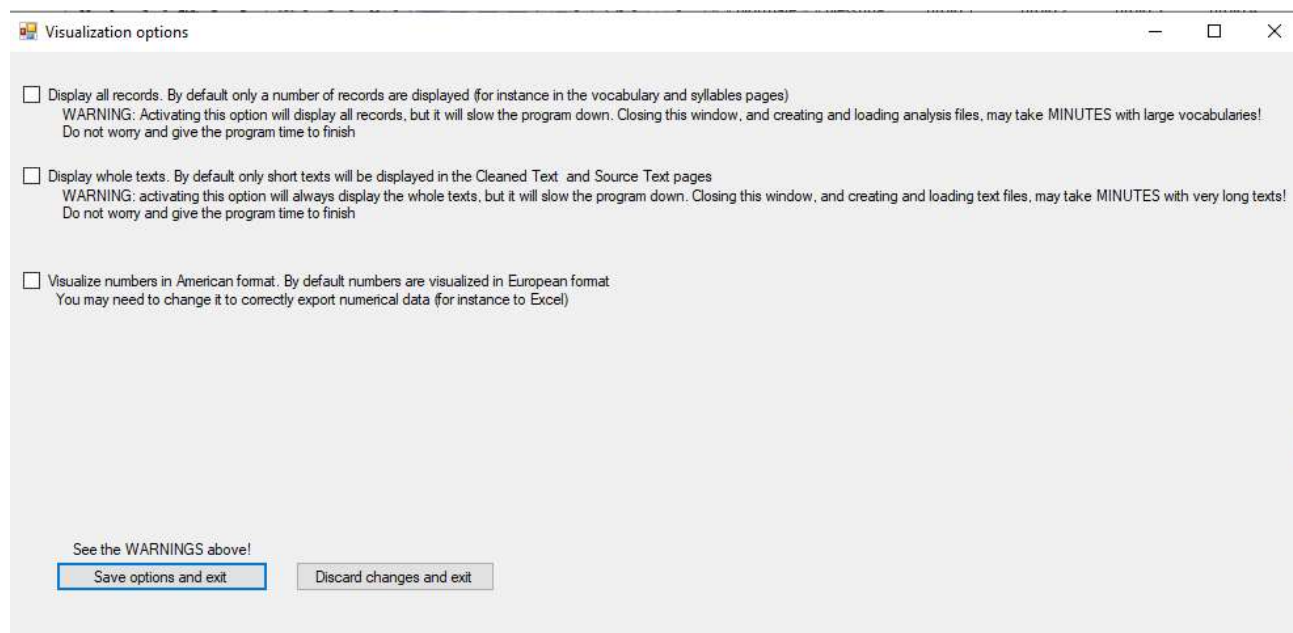
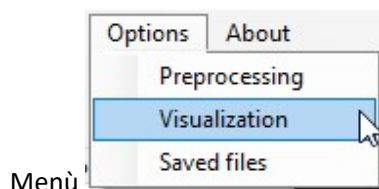
Nota: il messaggio è una sintesi di tutto quanto viene detto in questo paragrafo.

In questi casi consiglieri di usare <https://regex101.com>, copiandoci una parte del testo incriminato. Regex101 (fra le altre cose) visualizza i testi in un modo che consente di capire cosa siano effettivamente questi caratteri. Per esempio, questo è una parte del testo originale del Vangelo secondo Giovanni (di cui al messaggio più sopra) copiato in Regex101:



Da cui si vede che il carattere problematico, e altrimenti invisibile, è U+0092 (si chiama “Private use two”). Dato che TextAnalyzer lo sostituirà con uno spazio si vede anche che non causerà problemi nell’analisi, e si può evitare di correggerlo (cosa che richiederebbe una certa conoscenza di Regex, la sintassi da usare non è immediata come quella che si usa per sostituire un carattere ‘visibile’ (ma non è nemmeno molto complicata)).

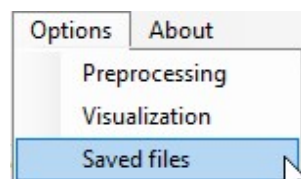
Opzioni di visualizzazione Analisi

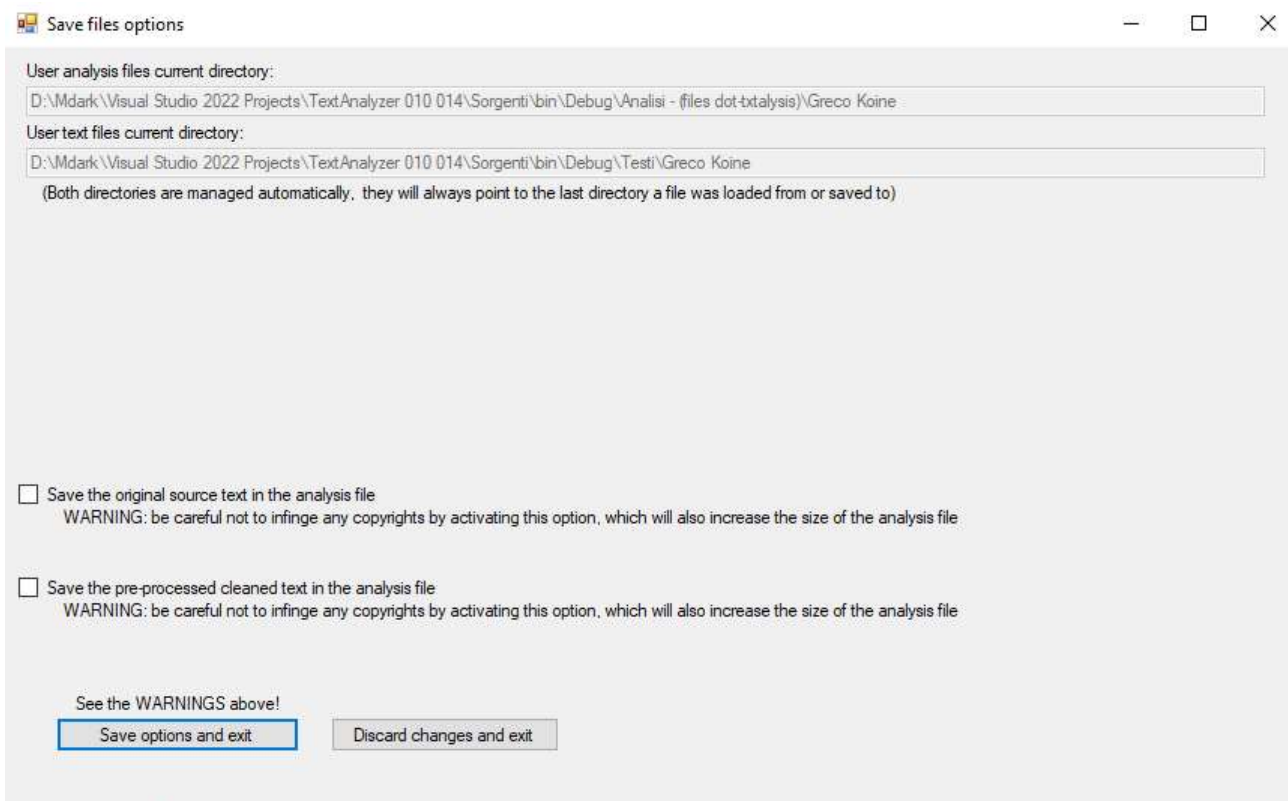


Le prime due opzioni servono per risparmiare tempo di esecuzione evitando di visualizzare il vocabolario completo (e le pseudo-sillabe) e i cleaned text/raw text. Le operazioni di scrittura su video sono per loro natura lente: un vocabolario di 300K parole può richiedere qualcosa come quindici minuti di tempo per essere visualizzato, quindi attivate queste opzioni con cautela e, se del caso, preparatevi da avere pazienza.

La terza opzione modifica il formato dei numeri da europeo (con la virgola decimale) ad americano (col punto decimale). Dovete settarla se, per esempio, il vostro Excel è settato per lavorare col formato americano (nel qual caso non 'capirebbe' i numeri scritti in formato europeo).

Opzioni di salvataggio Analisi (e copyrights)



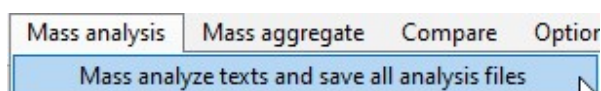


Da qua si può vedere quali sono le cartelle di lavoro di TextAnalyzer: per rendere il programma più maneggevole ce ne sono due, una per i files .txalysis e una per i files .txt. La posizione di entrambe le cartelle è gestita automaticamente da TextAnalyzer e punta sempre agli ultimi files a cui avete acceduto.

Le altre due opzioni abilitano il salvataggio nel file .txanalysis del testo originale e del cleaned text. ATTENZIONE: salvare una copia di un testo soggetto a copyright può essere considerato una violazione (nonostante il “fair use”). Inoltre salvare i testi serve a poco, quindi sconsiglio l’uso di queste due opzioni.

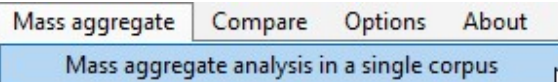
Bug conosciuto: evitare di aprire più finestre di opzioni contemporaneamente, perché premendo “save options” in una delle finestre vengono salvate anche tutte le altre opzioni e questo può facilmente generare una grossa confusione.

Funzioni di analisi di massa e aggregazione di massa



La funzione “Analisi di massa” consente di selezionare tanti testi quanti si vuole (devono essere tutti nella stessa cartella) e di analizzarli uno di seguito all’altro. Inoltre i files di analisi vengono automaticamente salvati, con lo stesso nome del testo originale (ma, ovviamente, con estensione .txalysis).

Nota: durante un’analisi di massa nella status window viene scritto solo un minimo di informazioni riguardo al processing di ogni file.



La funzione “Aggregazione di massa” consente di riunire quanti files .txanalysis si vuole in un'unica analisi, in modo da costruire un ‘corpus’ del linguaggio che può poi essere usato con le funzioni di comparazione (vedi paragrafo “Comparazione”).

Caricando un’analisi aggregata è possibile vedere, nella tab Report, con quali files è stata creata, oltre ad alcune informazioni relative ai files stessi. Questo è un esempio relativo ad un piccolo corpus Latino:

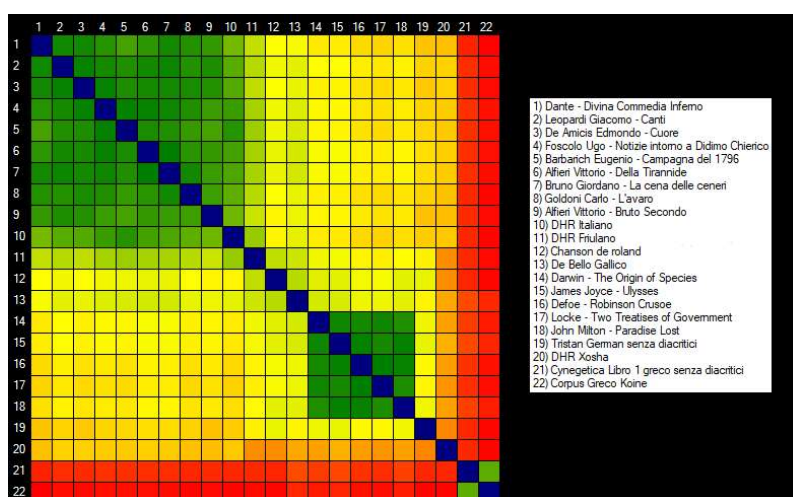
Files used in the analysis (10 files), with their raw length (if .txt) or cleaned length (if .txanalysis), character set size and maximum words length				
Alberti Leon Battista - Apologi centum.txanalysis	139090	25	18	
Alberti Leon Battista - De pictura.txanalysis	103664	25	18	
Alberti Leon Battista - Momus.txanalysis	296586	26	20	
Caesar - De Bello Gallico.txanalysis	369428	25	20	
Dante - De vulgari eloquentia.txanalysis	78480	33	27	
Dante - Monarchia.txanalysis	127342	28	19	
Dante - Quaestio de aqua et de terra.txanalysis	28878	25	18	
Tacito - Historiae.txanalysis	375448	26	20	
Tito Livio - Ab urbe condita.txanalysis	3477683	27	24	
Virgilio - Aeneis.txanalysis	431960	27	16	

Comparazione di testi

Con queste funzioni è possibile comparare varie analisi fra di loro. Per ognuna delle statistiche analizzate (distribuzione caratteri, bigrammi etc.) vengono calcolate le distanze geometriche (Euclidee) fra un’analisi e l’altra (con lo stesso metodo usato per calcolare le “Chars distances”).

Questo consente di:

1. Determinare in che lingua è stato scritto un testo, confrontandolo con testi in varie lingue. Questo è un esempio di quello che si ottiene:



Si vede bene dal grafico come i testi in Italiano (dall’ 1 al 10) siano tutti raggruppati nell’angolo in alto a sinistra, a poca distanza uno dall’altro (il colore verde). Anche i cinque testi in Inglese (14-15-16-17-18)

sono stati raggruppati assieme, e così pure un testo in Greco Classico (Cynegetica) viene raggruppato col Greco Koine, per quanto il colore verde più chiaro indichi che c'è una certa differenza.

Nota: i DHR sono le dichiarazioni dei diritti dell'uomo (Declaration of Human Rights). Lo Xosha è una lingua sud-africana: è distante da tutte le altre ma, dato che è scritta usando caratteri latini, non è così distante come i due testi scritti in caratteri greci.

L'identificazione automatica della lingua è una funzione molto affidabile.

Nota: può però fallire in casi limite. Per esempio, un testo scritto evitando appositamente l'uso di parole che contengono la lettera "e" avrebbe le sue statistiche alterate rispetto alla lingua originale e verrebbe probabilmente classificato come una lingua separata. Inoltre, serve un testo ragionevolmente lungo per ottenere dei buoni risultati (consiglierei di considerare che un testo sia 'ragionevolmente lungo' se ha almeno 11000 caratteri).

TextAnalyzer è in grado di identificare una lingua anche se il testo è stato crittografato usando un cifrario a sostituzione semplice (dopodiché è semplice ricavare una decodifica confrontando, per esempio, le tabelle dei bigrammi). Vedi paragrafo "Distanze blind e unblinded".

2. Comparando fra loro testi di una stessa lingua si può cercare di scoprire informazioni riguardo all'argomento trattato, o all'epoca in cui un testo è stato scritto, o al suo autore. Vedere nel seguito, e il paragrafo **"Errore. L'origine riferimento non è stata trovata."** per quello che si può fare (e per i limiti che un'analisi di questo genere ha).
3. Comparando un testo con un corpus della stessa lingua è possibile estrarre le parole comuni nel testo ma poco comuni nella lingua in generale. Anche questo consente di scoprire informazioni riguardo all'argomento, l'epoca, l'autore (o le idiosincrasie ortografiche dell'editore). Vedere nel seguito, e il paragrafo **"Errore. L'origine riferimento non è stata trovata."**

Distanze blind e unblinded

Le distanze fra un testo e l'altro vengono calcolate in due modi: "blind" (alla cieca) e "unblinded" (non alla cieca).

- Nel calcolo "blind" TextAnalyzer non fa alcuna supposizione su quello che i caratteri rappresentano nei vari testi. Le distanze vengono calcolate prendendo le tabelle così come sono, senza preoccuparsi, per esempio, se si stia comparando le statistiche della "a" in un testo con quelle della "e" in un altro.
- Nel calcolo "unblinded" TextAnalyzer suppone che un carattere rappresenti la stessa cosa in entrambi i testi, quindi le statistiche della "a" vengono comparate con le statistiche della "a", quelle della "e" con la "e" e così via.

Nota: questo vale anche per il vocabolario (e le pseudo-sillabe). Per esempio in un testo Italiano le parole più frequenti potrebbero essere "e" e "il", mentre in un altro testo potrebbero essere "il" e "la". In una comparazione blind la distanza viene calcolata sottraendo la frequenza con cui "il" compare nel secondo testo dalla frequenza con cui "e" compare nel primo, e sottraendo la frequenza con cui "il" compare nel secondo testo da quella con cui "il" compare nel primo. In una comparazione unblinded si usa la differenza fra le frequenze della "il" nei due testi e, dato che il "la" nel secondo testo non ha un equivalente nel primo, alla sua frequenza viene sottratto zero (e il numero risultante viene poi elevato al quadrato e sommato a quello precedente). La stessa cosa accade per la "e" nel primo testo. PS.: è più facile da capire che da spiegare xD.

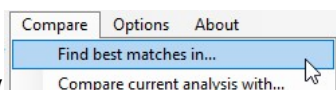
Il calcolo "unblinded" è più preciso e fornisce risultati più definiti, ma il calcolo "blind" è quello che consente a TextAnalyzer di riconoscere anche i testi crittografati.

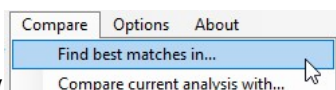
Nota: le statistiche del vocabolario possono essere "blind", ma ovviamente già il fatto di aver ricavato il vocabolario presuppone che il carattere 'spazio' sia effettivamente il separatore fra le parole.

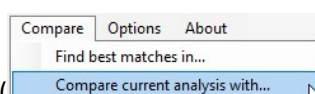
Nota: nel caso delle statistiche sulla lunghezza delle parole, ovviamente, il calcolo “blind” fornisce gli stessi risultati di quello “unblinded”.

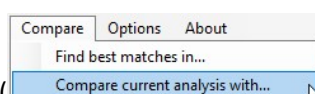
Funzioni di comparazione

Ci sono due modi per effettuare una comparazione: automatica e manuale.



In una comparazione automatica () vengono trovati, fra tutti quelli che vengono selezionati, i 24 testi più simili al testo corrente. Nota: al momento, “più simili” significa “con distanza minore, calcolata sulle tabelle di frequenza dei bigrammi”.



In una comparazione manuale () il testo corrente viene comparato con tutti quelli selezionati.

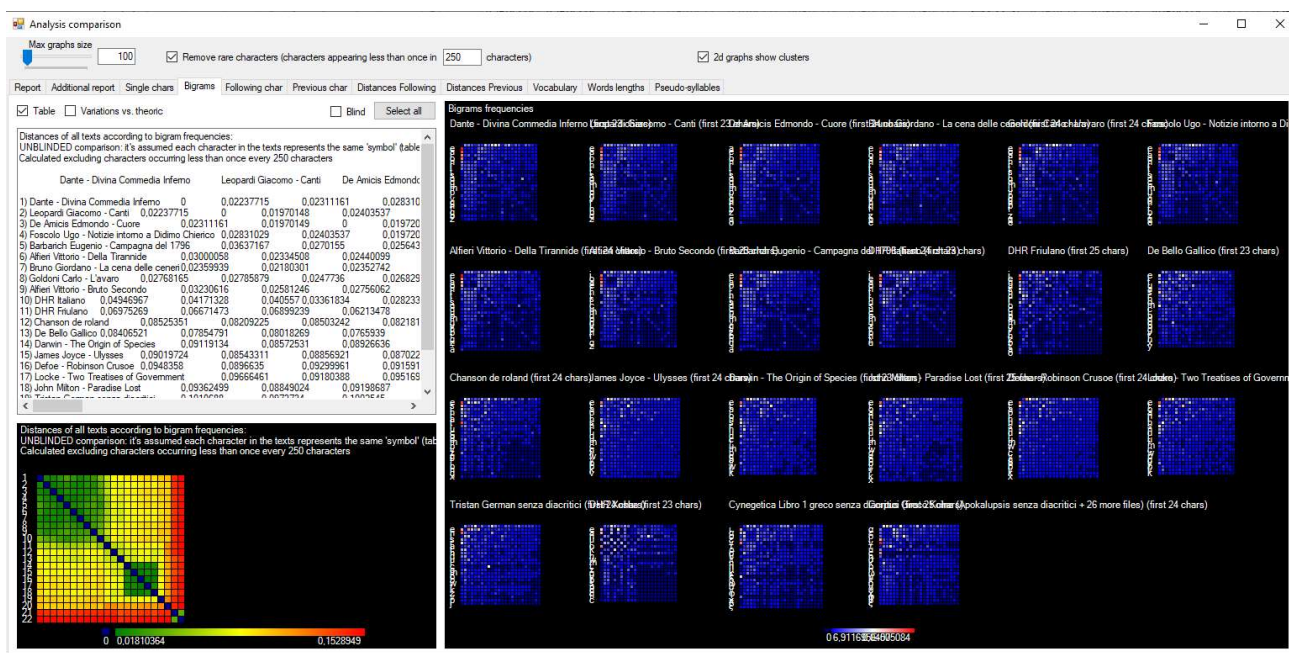
Nota: non c'è un limite sul numero di testi che possono essere comparati. Tuttavia, il tempo di elaborazione aumenta col quadrato dei numero testi: evitate di comparare numeri abnormi di testi, una cinquantina possono andare ancora bene, ma cento richiederebbero tempi di elaborazione lunghi. Inoltre molti dei grafici diventerebbero talmente ‘affollati’ da diventare indecifrabili (ammesso che ci sia abbastanza spazio sullo schermo per visualizzarli, il che non è detto nemmeno su schermi grandi).

Nota: se vengono comparate analisi eseguite con opzioni di preprocessing differenti viene dato un avviso.

Dopo una comparazione viene aperta una pagina apposita in cui vengono visualizzati risultati.

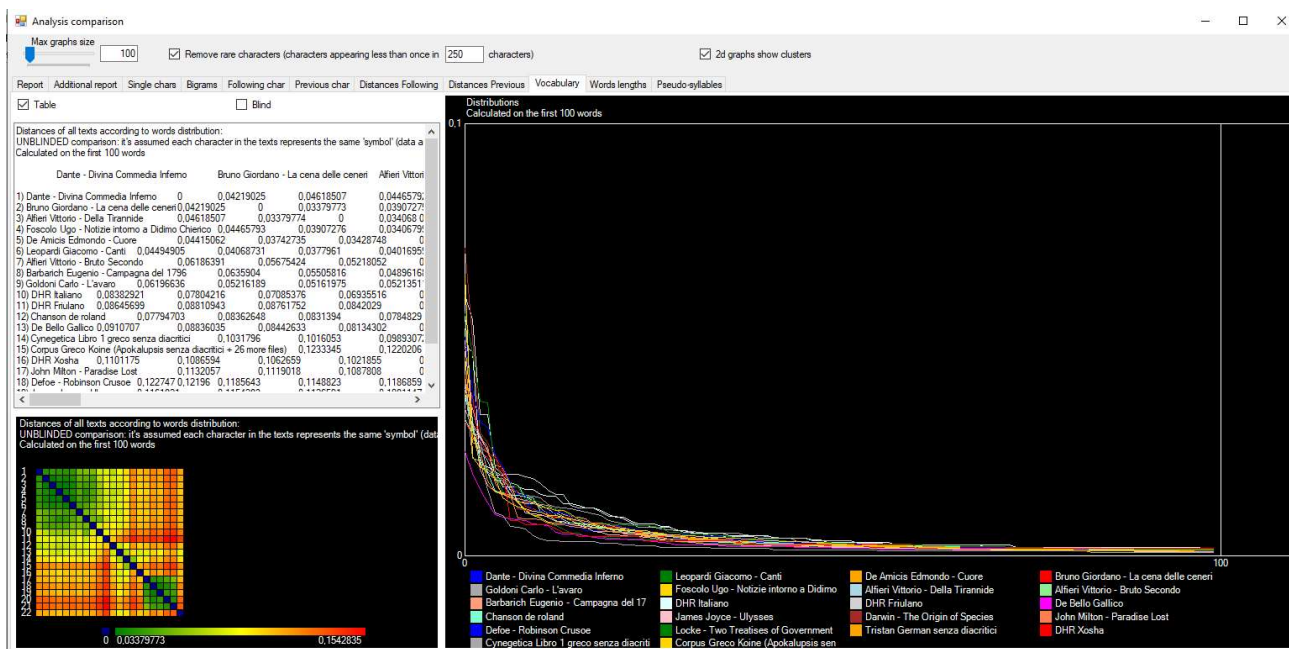
Pagina di comparazione

È organizzata in modo simile alla pagina di Analisi, con varie tabs che visualizzano le varie statistiche. Prendendo come esempio la pagina dei bigrammi (e la stessa comparazione della Divina Commedia – Inferno con testi in varie lingue vista precedentemente):



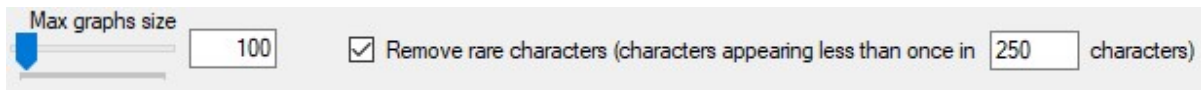
Sulla sinistra in alto troviamo la tabella con le distanze di ogni testo da un altro, sintetizzata poi nel grafico poco più in basso (il grafico di riepilogo). Sulla destra troviamo i grafici della distribuzione dei bigrammi di tutti i testi che vengono comparati.

Questo invece è un esempio con la pagina del Vocabolario, dove i grafici dei vari testi sono, ovviamente, monodimensionali (dato che vengono confrontate le frequenze delle parole, che sono delle liste e non delle tabelle):



Controlli nella pagina di comparazione

Sliders per settare i limiti dei grafici



The image shows a user interface for setting graph limits. On the left, there is a slider labeled 'Max graphs size' with a blue handle. To its right is a text input field containing the number '100'. Further right is a checked checkbox labeled 'Remove rare characters (characters appearing less than once in 250 characters)'. The number '250' is also in a text input field.

Anche nella pagina di comparazione è possibile scegliere i limiti dei grafici da visualizzare. Ci sono però due differenze importanti rispetto alla pagina di Analisi.

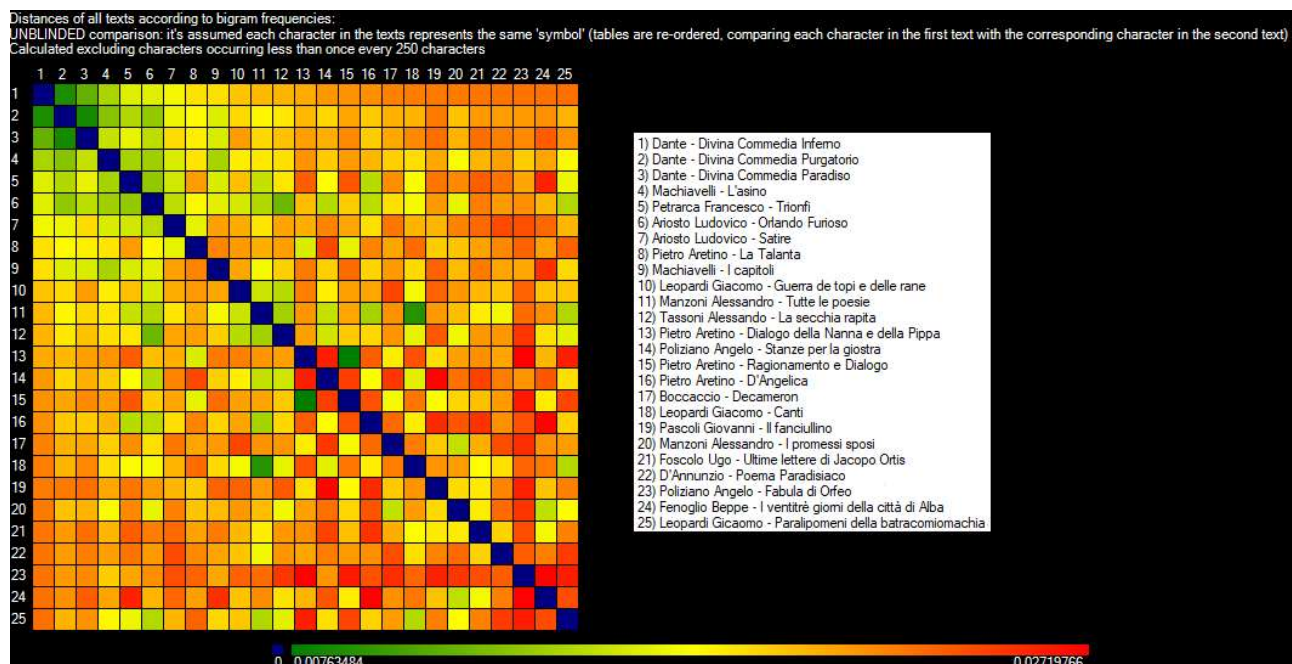
1. Mentre lo slider per i grafici lineari funziona normalmente, non esiste una slider per i grafici 2D. Questo perché ogni testo ha, in generale, una diversa distribuzione dei caratteri rari e, quindi, non è ragionevole settare un valore unico per tutti i testi. Per questo motivo c'è invece una casella numerica in cui si può settare il limite oltre il quale i caratteri vengono eliminati (TextAnalyzer calcola poi le dimensioni delle tabelle per ognuno dei testi).
2. Nella pagina di analisi gli sliders agiscono solo sui grafici. Qua, invece, i limiti *agiscono anche sui calcoli delle distanze*. Per esempio: se si setta un limite di 20 per la lunghezza dei grafici lineari il calcolo delle distanze dei vocabolari verrà eseguito solo sulle prime 10 parole. Questo consente di vedere come le distanze variano in funzione di cosa viene comparato.

Nota: da osservazioni preliminari le distanze “unblinded” fra testi di una stessa lingua diminuiscono se vengono considerati anche i caratteri rari (le distanze “unblinded” non cambiano significativamente).

Nota: dato che i limiti dei grafici agiscono anche sui calcoli, modificare questi valori richiede un certo tempo di elaborazione prima che vengano presentati i risultati. Questo tempo può diventare piuttosto lungo (anche svariati secondi) se si stanno comparando molte lingue contemporaneamente, specialmente se si setta un valore elevato per “Max graph size” o se si rimuove la spunta dalla casella “Remove rare characters”

Casella di spunta “2d summary graphs show clusters”

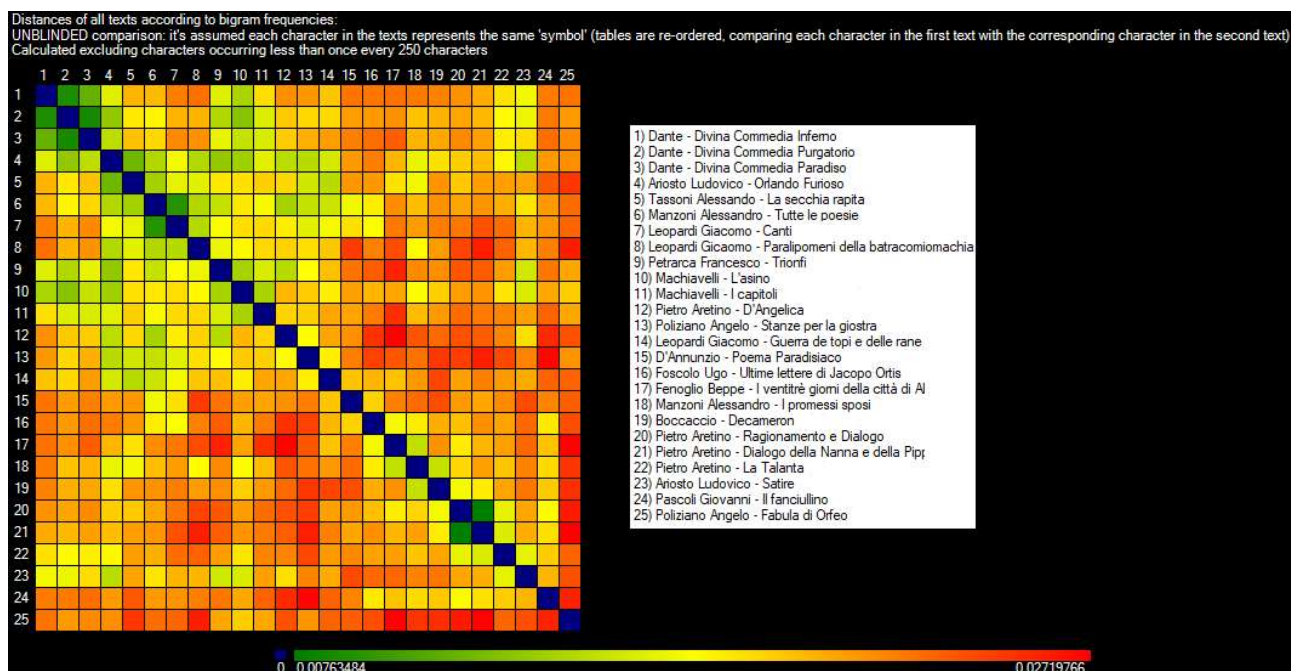
Di default la casella ☐ 2d summary graphs show clusters non è spuntata e i grafici vengo ordinati per distanze crescenti rispetto al testo che si sta comparando. Questo è un esempio dove la Divina Commedia – Inferno è stata confrontata (con una comparazione automatica) con altri 308 testi in Italiano scritti in varie epoche (dal 1300 agli anni 2000):



Con questo tipo di visualizzazione si possono apprezzare le distanze dal testo di riferimento agli altri. Dal grafico vediamo come le tre cantiche della Divina Commedia siano molto vicine fra di loro e come Dante, in generale, sia vicino a testi della sua epoca o a opere di poesia.

Nota: ricordate la scala dei colori è automatica ed è calcolata in modo tale da avere sempre la maggior risoluzione possibile. In questo grafico le differenze fra i testi sono piccole (al massimo ~ 0.027 , come si vede dalla scala in calce al grafico) e la scala colori le amplifica. Se, per esempio, confrontiamo questo grafico con quello del paragrafo "Funzioni di comparazione" vediamo che lì la distanza massima è molto più alta (~ 0.154 , 6 volte di più) e lì, quindi, le distanze di tutti i testi in italiano sono compresse in un blocco di quadratini colorati di verde, mentre qua sono espanse nei colori dal verde al rosso.

Quando la casella è spuntata, invece, i grafici di riepilogo 2D vengono ordinati in modo da evidenziare la presenza di gruppi di testi simili (è lo stesso tipo di visualizzazione già visto al paragrafo "Comparazione di testi", dove è stato usato per far vedere i vari gruppi di lingue):



Dato che in questo caso la scala colori è molto espansa e amplifica anche piccole differenze nei testi non si trovano molti gruppi, ma vediamo comunque che (oltre al raggruppamento delle cantiche della Divina Commedia) anche le poesie di Manzoni (il numero 6) sono vicine ai Canti di Leopardi (il numero 7), e così pure i due testi (20 e 21) di Pietro Aretino sono raggruppati fra loro..

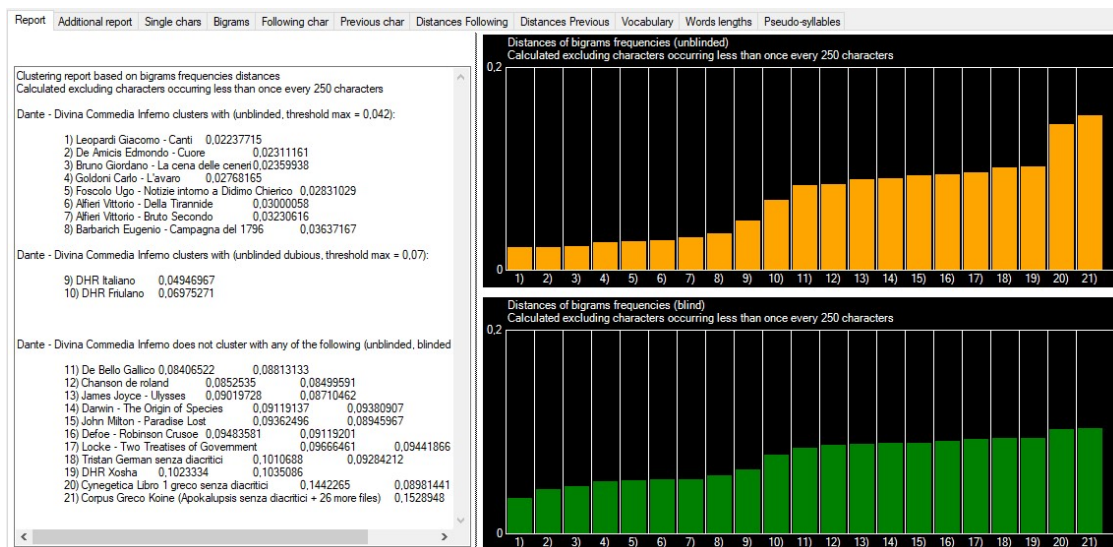
Nota: tenere a mente che le due modalità di visualizzazione non fanno altro che riordinare le righe e colonne delle tabelle, i dati numerici restano sempre gli stessi.

Nota: il riordinamento dei grafici di riepilogo per evidenziare i vari gruppi usa un algoritmo piuttosto semplice e potrebbe essere migliorato, ma è un lavoro complesso, e quello che c'è direi sia già sufficiente.

Nota: IMMAGINE CON CLUSTERING DA VECCHIA VERSIONE, ADESSO LA NUMERAZIONE SEGUE QUELLA DELL'ORDINAMENTO PER DISTANZE IN MODO DA NON TRARRE IN INGANNO

Report della comparazione

Questa tab è molto diversa dalle altre ed è particolarmente utile per il riconoscimento automatico della lingua, come in questo esempio (Divina Commedia – Inferno confrontata con testi in varie lingue):



Qua TextAnalyzer cerca di categorizzare automaticamente i vari testi in gruppi, considerando le distanze derivate dalla frequenza dei bigrammi.

Nota: vengono usate le distanze dei bigrammi perché sono quelle più affidabili per distinguere una lingua da un'altra. In futuro questa pagina potrebbe essere espansa per considerare anche le altre statistiche.

Come prima cosa vengono considerate le distanze unblinded, riunendo tutti i testi sufficientemente simili in un unico gruppo (che, in questo caso, è quello dei testi scritti in Italiano):

Dante - Divina Commedia Inferno clusters with (unblinded, threshold max = 0,042):

1) Leopardi Giacomo - Canti	0,02237715
2) De Amicis Edmondo - Cuore	0,02311161
3) Bruno Giordano - La cena delle ceneri	0,02359938
4) Goldoni Carlo - L'avaro	0,02768165
5) Foscolo Ugo - Notizie intorno a Didimo Chierico	0,02831029
6) Alfieri Vittorio - Della Tirannide	0,03000058
7) Alfieri Vittorio - Bruto Secondo	0,03230616
8) Barbarich Eugenio - Campagna del 1796	0,03637167

Nota: TextAnalyzer non usa un vero algoritmo di "clustering", usa semplicemente una soglia predefinita (in questo caso 0,042) per decidere a che gruppo assegnare un testo. Le soglie sono state determinate sulla base di prove preliminari e, almeno al momento, non sono modificabili.

Un'altra possibilità è che la distanza sia eccessiva per considerare un testo come appartenente alla stessa lingua, ma abbastanza bassa da lasciare un dubbio:

Dante - Divina Commedia Inferno clusters with (unblinded dubious, threshold max = 0,07):

9) DHR Italiano	0,04946967
10) DHR Friulano	0,06975271

In questo caso la dichiarazione dei diritti dell'uomo in Italiano è "dubbia" perché, essendo un testo piuttosto corto, è più probabile che le sue statistiche divergano da quelle degli altri testi in Italiano. Notare comunque che la sua distanza (~0,0495) è vicina alla soglia di 0,042 che l'avrebbe classificata come "Italiano". La dichiarazione dei diritti dell'uomo in Friulano (che è una lingua differente dall'Italiano!) rientra fra i "dubbi" per lo stesso motivo, ma anche qua notare come la sua distanza (~0,0698) sia molto vicina alla soglia di esclusione (0,07).

L'ultimo gruppo è quello dei testi che non hanno nulla a che fare con la Divina Commedia – Inferno, e qua troviamo tutti i testi non italiani:

Dante - Divina Commedia Inferno does not cluster with any of the following (unblinded, blinded distances):

11) De Bello Gallico	0,08406522	0,08813133	
12) Chanson de roland	0,0852535	0,08499591	
13) James Joyce - Ulysses	0,09019728	0,08710462	
14) Darwin - The Origin of Species	0,09119137	0,09380907	
15) John Milton - Paradise Lost	0,09362496	0,08945967	
16) Defoe - Robinson Crusoe	0,09483581	0,09119201	
17) Locke - Two Treatises of Government	0,09666461	0,09441866	
18) Tristan German senza diacritici	0,1010688	0,09284212	
19) DHR Xosha	0,1023334	0,1035086	
20) Cynegetica Libro 1 greco senza diacritici	0,1442265	0,08981441	
21) Corpus Greco Koine (Apokalupsis senza diacritici + 26 more files)	0,1528948	0,10;	

Per assegnare un testo a questo gruppo vengono usate sia la distanza unblinded che la blinded, questo perchè c'è ancora una possibilità: che il testo sia un crittogramma a sostituzione semplice, cosa di cui ci si può rendere conto tramite la sua distanza unblinded (come spiegato precedentemente) come in questo esempio:

Calculated excluding characters occurring less than once every 250 characters

Amore Loredana codificato inclusi spazi clusters with (unblinded, threshold max = 0,042): no other text

Amore Loredana codificato inclusi spazi is possibly a simple-substitution cipher, clusters with (blind, threshold max = 0,059):

15) De Amicis Edmondo - Cuore	0,03265668	
9) Leopardi Giacomo - Canti	0,03998813	
13) Alfieri Vittorio - Della Tirannide	0,0447384	
14) Barbarich Eugenio - Campagna del 1796	0,04563449	
10) Foscolo Ugo - Notizie intorno a Didimo Chierico	0,04655499	
18) Goldoni Carlo - L'avaro	0,0492719	
11) Bruno Giordano - La cena delle ceneri	0,05032373	
16) Alfieri Vittorio - Bruto Secondo	0,05623103	
20) DHR Italiano	0,05828154	

Amore Loredana codificato inclusi spazi does not cluster with any of the following (unblinded, blinded distances):

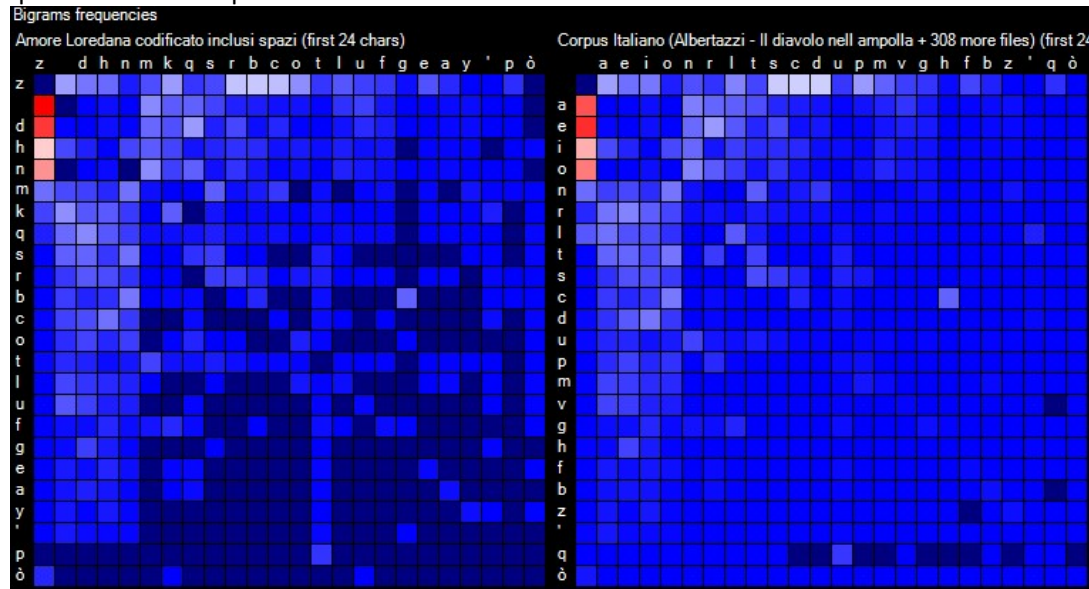
1) James Joyce - Ulysses	0,1311503	0,09072842
2) De Bello Gallico	0,1312518	0,0849515
3) John Milton - Paradise Lost	0,1330018	0,08667191
4) Darwin - The Origin of Species	0,1348474	0,09171226
5) Tristan German senza diacritici	0,1366694	0,09198583
6) Defoe - Robinson Crusoe	0,1368861	0,08945055
7) Locke - Two Treatises of Government	0,138257	0,09046682
8) Chanson de roland	0,1385143	0,08364438
12) DHR Xosha	0,138891	0,0975907
17) DHR Friulano	0,1401569	0,07831161
19) Cynegetica Libro 1 greco senza diacritici	0,1412438	0,08461873
21) Corpus Greco Koine (Apokalupsis senza diacritici + 26 more files)	0,1499771	0,09987194

Il testo "Amore di Loredana codificato inclusi spazi" è un romanzo di un autore (Luciano Zuccoli) di inizio '900, che è stato codificato (usano i comandi Regex di TextAnalyzer) sostituendo ogni carattere (incluso lo spazio) col successivo in ordine alfabetico, ottenendo un guazzabuglio di lettere:

*Inqdzchzknqdc m zcdkzldcdrhlnz tsnqdzk zbnlo fmh zcdkk zkdfddq zmnudkkdzkzk' Inqdzchzknqdc m zqnl
mynzchzktbh mnzyùbbnkhzlhk mnzeq sddkhhzsqdudrzdchsnqhzqnoqhdssàzkdssdq qh
zhzhqhshshzchzqhoqncityhnmddzdzchzsq ctyhnmddzrmnzqhrdqu shzodqzstssshzho drhzbnoqdrhzhk zrudyh
zk zmnqudfh zdzk'nk mc zlhk mnzshozsqdudrzk' Inqdzchzknqdc m zoqhl zo qsdzhzoqdmchzptdkkdu*

khfhdzdzonqs kdzhmzptdrsnzrbnlo qshldmsnztzoqdrsnzbgdzhkzsqdmnzqho qsdzk zunbdzmns
 zchdcdztmztrrtksnz zknqdc m zbgdzrs u zrnk z mbnq zbnkzudknzfghfnz

Dato però che le sostituzioni non alterano le statistiche dei bigrammi TextAnalyzer riconosce che si tratta di un testo Italiano tramite la distanza “blind”. A questo punto basta confrontare la tabella del testo codificato con quella dell’Italiano per trovare facilmente la decodifica:



E si vede facilmente che “spazio” è stato sostituito da “z”, la “a” da “spazio”, la “e” da “d” e così via.

Nota: sarebbe possibile anche suggerire automaticamente una possibile decodifica.

Unusual words

Comparando un testo con un corpus abbastanza esteso nella stessa lingua, cioè un corpus che rappresenti la lingua “media”, è possibile estrarre le parole comuni nel testo ma poco comuni nella lingua in generale. Questo consente di avere informazioni su vari aspetti:

- L’argomento del testo. Per esempio un testo che parla di medicina conterrà molti termini medici, mentre in un romanzo è probabile si trovino i nomi dei personaggi e delle località.
- L’uso da parte dell’autore di parole che possono far capire in quale epoca sia stato scritto un testo. Per esempio Dante usa frequentemente parole come “rispuose”, “ogne”, “sovrà”, mentre Salgari scrive “Spagnuolo” col dittongo “uo”, ormai obsoleto.
- L’uso particolarmente frequente di parole comuni da parte dell’autore, per esempio un uso eccessivo di “cioè” o “quantunque”. Questo potrebbe anche permettere di discriminare (coi dovuti *caveats!*) se è probabile che uno stesso autore abbia scritto due testi diversi.
- L’uso di grafie non-standard o l’esistenza di particolari idiosincrasie ortografiche dell’autore (o dell’editore) o anche di veri e propri errori ortografici (per esempio: parole scritte con l’accento sbagliato).

In questo esempio la Divina Commedia – Inferno viene confrontata con un corpus Italiano composto da 309 testi di varie epoche (Divina Commedia inclusa. Si tratta di circa 104 milioni di caratteri in totale e quasi 18 milioni di parole, con un vocabolario di 336990 parole):

Report	Unusual words	Single chars	Bigrams	Following char	Previous char	Distances Following	Distances Previous	Vocabulary	Words lengths	Pseudo-syllables
Search the first <input type="text" value="200"/> most frequent words Minimum amplification factor to accept a word as 'unusual': <input type="text" value="10"/>										
Words which are common in Dante - Divina Commedia Inferno, but are uncommon in Corpus Italiano (Albertazzi - Il diavolo nell ampolla + 308 more files)										
The first 200 most frequent words of Dante - Divina Commedia Inferno have been considered, with a minimum required amplification factor of 10. Take care: this function works best if the comparison is made with a large corpus, comparing against a small corpus or a single text will produce degraded results										
44 unusual words found:										
perche'	amplification factor =	547,1554	occurrences in text =	60						
che'	amplification factor =	428,2086	occurrences in text =	72						
giu	amplification factor =	396,838	occurrences in text =	66						
gia	amplification factor =	372,0657	occurrences in text =	102						
ch'e	amplification factor =	255,3392	occurrences in text =	21						
piu	amplification factor =	240,3765	occurrences in text =	181						
rispuose	amplification factor =	210,4444	occurrences in text =	30						
pero	amplification factor =	181,2235	occurrences in text =	52						
d'ogne	amplification factor =	159,587	occurrences in text =	21						
pie	amplification factor =	147,4603	occurrences in text =	38						
d'i	amplification factor =	141,6167	occurrences in text =	22						
ogne	amplification factor =	141,3485	occurrences in text =	31						
cio	amplification factor =	119,1583	occurrences in text =	49						
se'	amplification factor =	119,0031	occurrences in text =	92						
elli	amplification factor =	116,0633	occurrences in text =	42						
ch'i	amplification factor =	108,5957	occurrences in text =	78						
cosi	amplification factor =	94,79022	occurrences in text =	101						
inferno	amplification factor =	75,78842	occurrences in text =	41						
sovra	amplification factor =	65,30564	occurrences in text =	37						
i'	amplification factor =	59,05804	occurrences in text =	34						
com'io	amplification factor =	47,7865	occurrences in text =	20						
fummo	amplification factor =	41,45117	occurrences in text =	21						
sen	amplification factor =	38,04723	occurrences in text =	35						
allor	amplification factor =	32,18562	occurrences in text =	413						
l	amplification factor =	31,98064	occurrences in text =	38						
l'n	amplification factor =	31,93842	occurrences in text =	48						
loco	amplification factor =	31,15476	occurrences in text =	24						
convien	amplification factor =	29,57597	occurrences in text =	74						
ch'a	amplification factor =	23,78937	occurrences in text =	34						
vidi	amplification factor =	22,48168	occurrences in text =	47						
sanza	amplification factor =	21,1309	occurrences in text =	34						
l'un	amplification factor =	20,97326	occurrences in text =	45						
fui	amplification factor =	19,17601	occurrences in text =	54						
fuor	amplification factor =	19,02537	occurrences in text =	27						
poscia	amplification factor =	18,98868	occurrences in text =	29						
te'	amplification factor =	18,32276	occurrences in text =	20						
parea	amplification factor =	17,65018	occurrences in text =	30						
ciascun	amplification factor =	16,88751	occurrences in text =	78						
maestro	amplification factor =	14,6208	occurrences in text =	76						
ne'	amplification factor =	13,81522	occurrences in text =	87						
lor	amplification factor =	12,64343	occurrences in text =	61						
duca	amplification factor =	11,32174	occurrences in text =	58						
ancor	amplification factor =	10,6996	occurrences in text =	304						
li	amplification factor =	10,57305	occurrences in text =							

Sono evidenti le particolarità ortografiche dell'editore ("perchè" e "chè" scritti con un apostrofo invece che con un accento, "giu" e "piu" e "pie" non accentati), l'uso da parte di Dante di abbreviazioni poetiche ("ch'i", "i'", "com'io" eccetera) e di parole arcaiche ("rispuose", "d'ogne", "pie", "loco"...) e alcune parole correlate con l'argomento trattato ("inferno", "maestro", "duca").

Assieme alle parole trovate viene visualizzato il loro "fattore di amplificazione", cioè quante volte la parola è più frequente nel testo in esame rispetto alla lingua in generale, e il numero totale di volte che la parola compare nel testo.

La casella numerica Search the first most frequent words consente di regolare l'estensione della ricerca, di default limitata alle 200 parole più frequenti ma che può arrivare ad includere l'intero vocabolario (non c'è un limite massimo al valore che si può scrivere nella casella e la ricerca è sempre molto veloce), col che dal testo della Divina Commedia verrebbero recuperate anche parole molto caratteristiche come "draghignazzo", "flegetonte" eccetera).

Con la casella Minimum amplification factor to accept a word as 'unusual': è possibile regolare la sensibilità della ricerca.