UNIVERSITÀ DI PISA

CORSO DI LAUREA IN INFORMATICA UMANISTICA

Anno accademico 2012-2013

CHARACTERS AND AUTHORS: DIFFERENCES AND SIMILARITIES IN TERMS OF LANGUAGE. A STYLOMETRIC STUDY.

Candidato: Riccardo Galdieri

Relatore: Dott. Felice Dell'Orletta

Correlatore: Prof. Alessandro Lenci

27 Febbraio 2014

“Part of the inhumanity of the computer is that, once it is competently programmed and working smoothly, it is completely honest.”

Isaac Asimov, Change!, 1983

Contents

Abstract

What is the real difference between a fictional character and a real author, in terms of language? Is it possible to distinguish two fictional characters as if they were real, just relying on the book in which they feature? Would they have their own personal speaking style? Following the methods that (Khamelev and Tweedie, 2001) used ten years ago to answer similar questions, this project aims to apply stylometry techniques to fictional characters to see the effect of an algorithm based on Markov's chains when the subject is not a real author, but a fictional character that belongs to the world of imagination.

To do that, I created two pairs of corpora, the first containing all the texts written by two authors (Robert Louis Stevenson and Sir Arthur Conan Doyle), and the second containing all the sentences that belong to two fictional characters (Romeo and Macbeth); then I will test the same algorithms on both pairs of texts to see the differences produced within two separate types of domain. Specifically, I combined results obtained from 5 different classifiers that use Markov chains of order 0 and Markov chains of order 1 on different types of linguistic elements to determine the percentage of belonging of some sentences extracted randomly from a test set to the pair of corpora they are associated with.

1 - Introduction

The idea on which this project is based, the use of Markov chains to determine the percentage of similarity of a sentence within two corpora, needs to be considered as a development of the work made by Khmelev and Tweedie, who used Markov Chains on list of characters to attribute a sentence to a writer . In this case however, I am going to consider not only sequences of characters in my analysis, as they did, but also complex linguistic features extracted from different layers of the language such as Tokens, Parts Of Speech and the syntactic dependency relations.

In this first paragraph I will investigate the task and its background, taking a cue from similar tasks such as authorship attribution and sentence similarity; in the second paragraph then I will show the Corpora, their structure, and the methods I will use to make them; in the third paragraph I will show the features that I will extract with the preprocessing tools, and what these preprocessing tools means. In the fourth paragraph I will show the mechanism that I will use to evaluate the system in analysis, including an explanation of the classifiers and their features. In the fifth paragraph I will run a pilot test, in order to measure the behaviour of some very important

linguistic elements in context; in the sixth paragraph, I will show the full analysis and the results, with another discussion concerning the data, the differences between what I got and what I expected to have, and I will make a global discussion about the reliability of the response. I will lastly expose my general conclusions and some ideas for future developments.

1.1 Previous Works

The development of this kind of work can be considered relatively recent in Computational Linguistics, for several reasons. The first and most significant one is that only in the last years computers and electronic devices have reached the potential to process big amounts of data, making large-scale analysis accessible to all personal computers; also the difficulty in isolating the “knowledge” of a fictional character has been a deterrent to this type of investigation so far. In the last years however, some researchers have laid the groundwork for further advancements: (Bethard et al., 2004) proposed a method to associate opinion propositions with their holders, (Elson and McKeown, 2010) used supervised Machine Learning algorithms to associate 3,000 quotations with the right speaker, choosing from a set of six authors, and (Hua, et al., 2013), who tried to identify speakers in novels, again using a supervised machine learning approach.

All these works can be catalogued as “speaker recognition” tasks, and they might be useful because they could potentially produce better and richer materials to be applied in tasks like mine. Unfortunately however, they are not revealing so far as the possible association between texts and authors is concerned.

These preliminaries must be taken into consideration to account for the fewness of endeavours of this kind. The most similar work was made by (Brooks et al., 2013), who attempted to cluster the many voices in 'The Waste Land'; the paper could be considered a work on “speaker recognition”, the many voices were first segmented, then each of them has been profiled and analysed to identify the speaker.

In order to give an exhaustive explanation of this work, all the elements which were helpful in each single step of my work must be considered individually: stylometry, sentence similarity and authorship attribution.

1.2 Stylometry

Stylography, also known as **linguistic profiling**, is the statistical analysis of variations in literary style among writers or genres. Unlike the speaker recognition, stylometry is a very old

task: the first attempt was made by the English logician Augustus De Morgan, who suggested in a letter to a friend, that the length of words could be an indicator of authorship (De Morgan, 1882). As reported by (Holmes, 1998) in his paper "The Evolution of Stylometry in Humanities Scholarship" , De Morgan's idea was investigated by Thomas Mandenhall, who subsequently published the results of his labours in measuring the lengths of several hundred thousand words from the *ouvre* of Bacon, Marlowe, and Shakespeare (Mendenhall, 1887). Just three years later, the Polish philosopher Wincenty Lutoslawski published the book *Principes de stylométrie*, in which he applied the same method to build a chronology of Plato's Dialogues.

During the following decades, the approach to stylometry remained the same, and the only change introduced was the type of feature that researchers tried to extract. (Yule, 1938; Williams, 1940, Cox and Brandwood, 1959). Unfortunately the results were for a long time not quite satisfying. The first demonstration of good use of stylometry techniques is dated 1964, when the American researchers Mosteller and Wallace successfully employed function words such as prepositions, conjunctions, and articles to determine the writer of contested federalist papers[1]. Unfortunately, as noticed by (Ramyaa and Rasheed, 2004), features can not be generalized:

> "There has not been a comparison of results on a large scale as to what features are generally more representative or what methods are typically more effective. [...] Particular words may be used for a specific classification (like The Federalist Papers) but they cannot be counted on for style analysis in general. "

Nowadays stylometry has become a very useful investigation method, and it is also used to detect improper interventions on text, such as vandalism (Harpalani et al, 2011) and promotional contents (Bhosale et al., 2013) in Wikipedia, or to extract informations such as age, gender, native language, type of instruction, and many other similar features about the writer of the analysed text (van Haltaren, 2004; Argamon et al, 2008).

1.3 Sentence Similarity

In computational linguistics one of the most well known tasks is the **sentence similarity**. Since the first studies carried out by (Chomsky, 1957) in the late 1950's, the problem of "what can be considered similar to what" has been studied in depth by scholars who have focused on the

1 The Federalist Papers are 85 articles written by Alexander Hamilton, James Madison, and John Jay to convince the New York State to ratify the American Constitution. Some of these articles however were a matter of contention among the writers because they were not signed. Thanks to this ambiguity, they have been used as test case for a lot of stylometric studies.

definition of “similar” to determine which criteria have to be applied to solve the issue. As shown by Chomsky himself, language is not a mere concatenation of words the meaning of which depends on the previous and the following elements; in fact, the structure of language itself has a crucial role for the meaning-making of a sentence; some words may depend on another positioned further on in the statement. This idea has been the input for subsequent developments of “context free grammars” and its derivation trees (Tatcher, 1967), which are now used to describe languages in terms of structure. Unfortunately, however, the structure of a sentence does not fully explain the mechanism used to generate a statement; it is possible to have two different sentences with the same structure and with a good percentage of words in common, but with opposite meanings. The obvious conclusion to this is that structures have to combine words position with something else. (Bar-Hillel, 1954) has been the first to place attention on the relationship between syntax and another sentence's constituent (Owens Jr, 1984): the semantics. In the next thirty years the debate about this relationship has been very popular throughout the linguist community, especially regarding whether the element which is most to be valued for the purposes of a linguistic analysis is the syntax or the semantics. In late 80's, more papers started to demonstrate how a solution could combine these two values to avoid favouring one over the other. It is appropriate in this regard to report (Lytinen, 1996) introduction to his paper about semantics and natural language processing:

> “Over the last decade, many researchers in Natural Language Processing (NLP) have begun to realize that semantics and pragmatic information must have more influence on parsing. The reason or this is that many syntactic ambiguities cannot be resolved without reference to semantics or pragmatics”

Another attempt at conciliation was made by (Lytinen, 1986) in a paper which deals with the combination of syntax and semantics as equal members, ignoring any idea of priority of the first or the second element. Nowadays, every application of sentence similarity methods cannot disregard this dualism, as shown in the article “Sentence Similarity Based on Semantic Nets and Corpus Statistics” by (Li, et al. 2006). Li's idea was to calculate two different values, the first one using the words order and incorporating it into the sentences; the second, of course, was to use semantic database strategies to determine how the “meaning” of the inputs were similar. Combining these two values, Li's algorithm gave its final output.

1.4 Authorship Attribution

The authorship attribution task includes all those branches of Computational Linguistics that study the methods of association between authors or characters and their styles. This topic is very old in this field and the first attempts to face it date back to 1887 (Mendenhall, 1887). What has always been considered as important, even before linguistic features were focused on, was the dimension of the corpora and the power of calculators used to perform any analysis. It was unthinkable at the beginning of the 20th century to solve automatically tasks on big dimensions of data, due to problems with the physical capacity of a computer to handle that information. The only alternative was to codify by hand the texts, but it required long time and a high number of people at work. Just to make a comparison, the first attempt to create automatically a corpus of significant dimensions was made at the Brown University in 1961: the corpus was called Brown University Standard Corpus of Present-Day American English (or just Brown Corpus), and it counted around a million words (Francis and Kucera, 1979). Nowadays, a corpus of big dimensions, like the Corpus of Contemporary American English, might count more than 400 millions of words. (Davies, 2009)

The importance of the calculation capacity of computers was evidenced by (Holmes, 1994), who pointed out the importance of technological developments in order to perform statistical analysis . A practical example was supplied by (Hepple, 2000) some years later, when he explained how POS-tagger's script were relevant for a good computing. Going back just one step, at the beginning of the modern authorship attribution history, it is worth mentioning some milestones of this field: (Mosteller and Wallace, 1964) tried to identify the authors of some federalist articles of 1778 applying the Bayes' theorem to the shortest words of the texts. This work, as evidenced by (Stamatatos, 2009) “initiated non-traditional authorship attribution studies, as opposed to traditional human expert-based methods”, in other words, stylometry(Holmes, 1998). One of the techniques applied in this new branch of Computational Linguistics was to apply Markov chains to understand the relationship between the order of words and the authors' style (Juola, 1998; Juola and Baayen, 2005; Khmelev, 2001). In this context I can mention the first paper quoted in this section from Khmelev and Tweedie. As I said, however, I will not only consider concatenation of words, but following in the footsteps of several previous studies (Chaski,2005; Argamon and Levitan, 2005; Koppel, Argamon and Shimoni, 2002). I am going to look at how different linguistic features influence the style of a writer.

2 - Corpora

To perform the analysis required by this task, I needed to create two pairs of corpora. The first is composed by two collections of sentences pronounced by two fictional characters, with an order of magnitude of 300 sentences, the second by texts from two different authors with an order of magnitude of 25000 sentences. The reason is pretty much obvious: considering the fact that no one has ever analysed characters corpus (CC) before, I needed other elements to be used during the analysis, that could assure me of having made no mistakes with my code, and that could work as a term of comparison for my results as well. Of course these analysis are representative of two different tasks, with this project I want to see if the same approach can be used whether looking at characters or authors.

2.1 Characters Corpus

Considering the task of extracting the dialogues from a given text too complex and long for the scope of this project, I searched for a corpus that could be readily used. An appropriate solution was provided by the web-page of "The Complete Works of William Shakespeare" on the MIT[2] server. This project, started in 1993 by Jeremy Hylton[3] with the purpose of collecting all Shakespeare's productions, does not have a list of sources or a precise indication of where the texts were taken from; only when a work on that page is selected, there appears a link to the Amazon Online Bookshop, advertising that a copy of that text is available on it. The correlation between the texts and the respective books is only supposed then and the authority given by the MIT's domain has been the determining factor that persuaded me choose it, albeit with some reservations. Specifically, I took all the phrases pronounced by Romeo from *Romeo and Juliet*, and by Macbeth from the play *Macbeth*. Unfortunately, the pages code was not structured with a precise relationship between the speaker and the sentences he says; the element containing the character's name was a child of a containing element and was in turn the sibling of a couple tags that contained the text I intended to collect. The structure was similar to the following:

```
<A NAME=speech3><b>LORD POLONIUS</b></a>
<blockquote>
        <A NAME=3>You shall do marvellous wisely, good Reynaldo,</A><br>
        <A NAME=4>Before you visit him, to make inquire</A><br>
        <A NAME=5>Of his behavior.</A><br>
</blockquote>
```

Tab.. 2.1.1 – Webpages structure sample

2 Massachusetts's Institute of Technology

3 The website's home page URL is http://shakespeare.mit.edu/

The best solution in this case was to find something that allowed me to navigate the page structure as quickly as possible. The solution I have chosen was to use BeautifoulSoup, an Object-Oriented Python library created to explore, read, and extract information from HTML and XML files. I have created an algorithm that for each character, was able to read the source, find the character's name, create a list of children containing the text for which we are looking, and put all these sentences into an output file of raw utf-8 text. The extracted phrases needed some other little fixes, like deleting the line breaks and some special characters, but the output was clear and correct. I also decided to create two copies of CC with lowercased text that could be used after the pilot test.

The punctuation nonetheless was neither removed nor changed: as a matter of fact, it is crucial in every sentence to understand the real meaning, so that changing the punctuation would also change the meaning of the sentence itself. For example, the two sentences “I ate, my cousin” and “I ate my cousin” are similar in terms of words, but not in terms of meaning. This difference is also very important for the syntactic parsing, that would not split sentences correctly without taking it into account.

2.2 Authors Corpus

A different approach was required to create the authors corpus (AC). Following Mercer's idea that “more data is better data” (Church and Mercer, 1993), I decided to collect around 500.000 words (not tokens) for each corpus getting closer to the Brown's corpus dimension. A huge source of free texts available on the web is offered by the Project Gutenberg, which was created in 1971 by Michael Hart as first source for eBooks in the world (Lebert, 2008). As reported by the site's home page, the project offers a big collection of texts in several extensions (such as HTML, raw text, PDF and sometimes audio books). Even if they specify that the texts on the website are the digitalized version of real books, users sometimes expressed doubts about the quality of these. In this case however, I decided to consider the website a trusted source and to use its texts.

In a first attempt, I compared a collection of texts from Robert Louis Stevenson with another collection from Charles Dickens. The number of words counted with a simple plugin for Sublime Text Editor gave 526.101 words for Dickens, 489.552 for Stevenson. The difference was not significant enough to cause concern, but I still decided to search for some other element which would help reducing the gap. Considering that it could not have been easy to create a collection of

texts with precise numbers of words, I tried with a collection of short stories and novels in a manner that could have been able to reach a precise number of words that added texts one by one. The result has been relatively positive as regards Sir Arthur Conan Doyle: the corpus created using his stories is of 506.639 words, which is close enough to the Dickens one. In this second creation of corpora, I did not use any automatic tool to grab texts. The number of these and the number of controls required during the process suggested the creation of a corpus with a better control on the dimensions just by copying and pasting their utf-8 version into a single document.

The list of texts converged in the corpus are, for Sir Arthur Conan Doyle:

- A study in Scarlet
- The Adventure of Wisteria Lodge
- The Adventure of the Cardboard Box
- The Adventure of the Devil's Foot
- The Adventure of the Dying Detective
- The Adventure of the Red Circle
- The Disappearance of Lady Frances Carfax
- Memoirs of Sherlock Holmes
- The Return of Sherlock Holmes
- The Sign of the Four
- The Adventure of the Bruce-Partington Plans
- The Adventures of Sherlock Holmes
- The Hound of the Baskervilles
- The Valley of Fear

and for Stevenson:

- Treasure island
- In the South Seas
- The Wrecker
- The strange case of Dr. Jeckyll and Mr. Hyde
- The black arrow
- Master of Ballantrae

Also in this case some small corrections were required, like removing the chapter's list, or a small footer inserted by the website. None of the corpora has been codified, so as to quicken the POS-Tagging phase and to make the “text reading” easier for NLTK.

3 – Preprocessing Tools: Linguistic Analysis

In this study, I focused on two different sets of features: the first, regarding the corpora, contains a sequence of indexes I used to make sure that both pairs of corpora were similar; the second, regarding the sentences extraction, is used to create the five classifiers that I will use to test the accuracy of the system.

All corpora were automatically morphosyntactically tagged by the POS tagger described in (Dell’Orletta, 2009) and dependency–parsed by the DeSR parser (Attardi, 2006) using Support Vector Machine as learning algorithm. DeSR, trained on the ISST–TANL treebank consisting of articles from newspapers and periodicals, achieves a performance of 83.38% and 87.71% in terms of LAS and UAS respectively when tested on texts of the same type (Attardi et al., 2009).

All algorithms unfortunately have suffered domain adaptations problems. When a linguistic tool is tested, the result is often the accuracy percentage, but when the corpus on which the system has been tested is changed, the accuracy is drastically reduced (Gildea, 2001). The AC contain texts written by authors who lived at the end of the 19th century, hence they rely on a different linguistic style from the WSJ corpus, that is usually used to test linguistic tools. The CC instead contain texts that are supposed to be transcriptions of speeches of characters who lived in the 16th century, and that therefore used a different vocabulary.

3.1 Linguistic profile

After this small digression, I am now going to list the indexes that I have extracted from each corpus:

Number of sentences: The number of sentences per corpus.

Number of tokens: The number of tokens per corpus.

Number of characters: The number of characters per corpus.

Number of hapaxes: The number of tokens occurring only once in the corpus.

Hapaxes distribution: The ratio between the number of hapaxes and tokens.

Sentence Length average: calculated as the average number of characters per sentence.

Tokens Length average: calculated as the average number of characters per word.

Vocabulary size: The number of distinct tokens in the corpus.

Grammatical Tokens Percentage: The percentage of grammatical words in the texts. I have listed as grammatical words symbols, interjections, conjunctions, possessive pronouns, prepositions and punctuation.[4]

Type/Token Ratio: the Type/Token Ratio (TTR) is a measure of vocabulary variation which has shown to be helpful for measuring lexical variety within a text.

Lexical density: it refers to the ratio of content words (verbs, nouns, adjectives and adverbs) to the total number of lexical tokens in a text.

Named Entity: The named people or places.

"I" Occurrences: Considering that I was investigating on fictional characters, the occurrences of I as subject of the sentences, the percentage of "I" as subject in the whole corpus, and the distance of the subject from the sentence's head could be considered as relevant features.

3.2 Preprocessing tools results

This first step, extracting and comparing the first set of features from both pairs of training set, gave me an idea of how similar the corpora were**.** These are the results for lower and uppercased CC:

	Romeo	**Macbeth**
Number of phrases:	216	196

4 Following the Treebank notation, I have listed as grammatical words EX,LS, SYM, UH,CC, PRP, PRP$, IN, DT, TO, POS and punctuation.

Number of tokens:	4764	5661
Number of characters:	16702	20296
Number of hapax:	780	984
Hapax's Distribution:	0.163727959698	0.173820879703
word's length average:	3.50587741394	3.58523229111
Vocabulary's size:	1209	1458
Grammatical Words%:	46.9799825936	47.9790188214
Type Token Ratio:	0.253778337531	0.257551669316
Lexical Density:	0.595654797827	0.56703026038

3.1 – Preliminary indexes on the lowercased Characters Corpus.

	Romeo	**Macbeth**
Number of sentences:	297	310
Number of tokens:	5227	5844
Number of characters:	18147	20644
Number of hapax:	854	1062
Hapaxes Distribution:	0.163382437345	0.181724845996
Sentences length average:	74.3636363636	80.9064516129
Tokens length average:	3.47178113641	3.5325119781
Vocabulary's size:	1365	1595
Grammatical Tokens%:	45.8137510879	46.8764460898
Type Token Ratio:	0.26114405969	0.272929500342
Lexical Density:	0.609535304768	0.582336382829

3.2 – Preliminary indexes on the uppercased Characters Corpus.

The differences highlighted in tables **4.1** and **4.2** are significant. The lowercased version of Macbeth counts more than a hundred sentences less than the uppercased version, one sentence on three being merged with another sentence. Also the number of hapax is significantly changed. All the other indexes, more or less, follow the same order of magnitude; every variation, when not significant, can be attributed to the sentences that the program has put aside for the test.

Besides these data, I tried to extract other informations from the corpora, specifically the Named Entity (people and places) and some statistics about the role of the subject. With no surprise, NLTK's NER did not find any entity in the lowercased corpora, a terrible score compared with the hundred and more that were found inside the other corpora. About the subject however, the

analysis has given similar result in both the analysis (I am showing these results below in this paper). In any case, the reliability of lowercased corpora is roughly zero, so henceforth I will consider only uppercased corpora as representative of CC.

Let's see now how results changed for the AC:

	Doyle	**Stevenson**
Number of sentences:	23762	18634
Number of tokens:	569281	544221
Number of characters:	2102765	2012940
Number of hapax:	10715	14152
Hapaxes Distribution:	0.0188219877354	0.0260041416998
Sentences length average:	108.648598603	131.927659118
Tokens length average:	3.69372067573	3.69875473383
Vocabulary's size:	22902	28237
Grammatical Tokens%:	48.2683585834	50.3061603709
Type Token Ratio:	0.0402296932446	0.051885171649
Lexical Density:	0.545024788949	0.530360546563

3.3 – Preliminary indexes on the Authors Corpora.

In this occasion, too, the statistics have changed. Two thousand sentences are kept aside, and the different sentence-splitter used to create the CoNLL file changed the quantity of remaining elements. Luckily, the difference with table **X.X,** in terms of percentage, is not significant. Looking at the distribution of named places, I noticed something interesting in Doyle's corpus.

1) London 274
2) Holmes 114
3) England 99
4) English 50
5) America 41

If the presence of English could be forgiven, “Holmes” in second position cannot be commented on.

4 – Classifiers

It is useful in this case to remark that I have used the 90% of the whole size as training set, and the remaining 10% as test set..

4.1 The approach

The analysis is particularly complex, and needs an example to be fully explained. Let's take the CC as sample. The algorithm takes Romeo's test set, and for each sentence in it, uses some classifiers to establish a value of belonging to both Romeo and Macbeth training set. If the belonging value is considered more significant in Romeo's training set, a counter is increased. Each sentence passes through ten classifiers, specifically five classifiers using Markov chains of order 0 on Characters, Tokens, POS, SDR and a last classifier that multiplies the other four, and five other classifiers that applies the same system using Markov chains of order 1[5]. The same operation is repeated using the Macbeth's test set. After these analyses I obtained the percentage in which the classifiers attribute the sentence to the right corpus. For each group of classifiers I also inserted a voting system: if more than the 50% of classifiers successfully assigned the sentence to the right training set, a counter was increased. This classifier will be called Local Voting System. For each classifier, the score accuracy has been evaluated by dividing the number of assignments with the test set size.

One last classifier is used, it is called system accuracy and works like the Local Voting System, but it takes into account both order 0 and order 1 classifiers, increasing its value when at least six classifiers on ten have made a right assignment. This classifier will be called Global Voting System

4.2 Linguistic features of the classifiers

Unlike the first set, which is applied once and for the whole corpora, the second set I worked with is made of features which were each extracted twice for every test sentence. For each feature listed below, I used two classifiers, one using Markov chains of order 0, and one using Markov chains of order 1. The sentences added to the test set were randomly chosen from the corpus they belong to. These are the features extracted from each sentence:

5 A Markov chain isia mathematical system that undergoes transitions from one state to another. It is a random process usually characterized as memoryless: the next state depends only on the current state and not on the sequence of events that preceded it.

Characters: trivially, the list of characters (including punctuation and excluding spaces).

Tokens: the list of tokens composing the sentence analysed.

Part Of Speech (POS): for each token, the corresponding part of speech.

Syntactc Dependency Relation (SDR): Dependency relation to the head of the sentence.

5 – Pilot test

Before performing the real analysis, I did some experiments. I reduced the number of classifiers, using only POS and tokens, and no division between training set and test set has been done, sentences were extracted directly from the corpora. This test is motivated by some doubts I had about the behaviour of some parameters during the extraction, such as Missing occurrences and the lowercased corpora.

5.1 First test and discussion

The program has been executed twice for each pair of texts, at first extracting phrases totally random, the second time having a minimum of 30 characters for each sentence extracted. This measure has been taken to stem short sentences like “Not yet” or “That's true”, considered as usually not relevant to determine the style of an author. The number 30 has been based on the average of words for the long texts' vocabulary (around 7 as show in Tab. 2.2); that means that we needed to extract sentences with at least 4 words, or a very long word that would be relevant in accordance with Zipf's Law (Li, 1992).

An important decision I had to take was about the number of missing occurrences that could appear in our process. If you extract a sentence from Romeo, for example, it might be possible to find the token “Juliet”. A missing occurrence of this value during the comparison with Macbeth would nullify the value of the whole chain. The decision in this case has been to ignore these values, considering the amount of data big enough to reduce the number of missing occurrences to a very small value.

T

hese are the results I obtained extracting 80 sentences from the CC:

	Fully Random Phrases	Controlled Length Phrases
From the first text:		
Order 0 Chains(tokens)	11.25%	8.75 %
Order 0 Chains(POS)	73.75 %	70 %
Order 1 Chains(tokens)	16.25%	7.5 %
Order 1 Chains(POS)	63.75%	56.25 %
From the second text:		
Order 0 Chains(tokens)	12.5 %	1.25 %
Order 0 Chains(POS)	47.5 %	56.25 %
Order 1 Chains(tokens)	18.75 %	16.25 %
Order 1 Chains(POS)	48.75 %	48.75 %

Tab 5.1 – Characters corpora results

and these are the results extracting 250 sentences from the AC:

	Fully Random Phrases	Controlled Length Phrases
From the first text:		
Order 0 Chains(tokens)	49.6%	50%
Order 0 Chains(POS)	72.8%	60.8%
Order 1 Chains(tokens)	16.4%	12%
Order 1 Chains(POS)	82.8%	78.4%
From the second text:		
Order 0 Chains(tokens)	18.8%	20.4%
Order 0 Chains(POS)	61.6%	74.4%
Order 1 Chains(tokens)	15.6%	18.8%

Order 1 Chains(POS)	59.2%	62.8%

Tab 5.2 – AC results

Looking at the results, it is clear that the principle does not work with the tokens in these texts. There is no significant difference between the values for order 0 and 1 chains which are usually between 0.1 and 0.15. It means that not only is the order of tokens not relevant for our purposes, but also that in more than the 80% of the cases the order is associated to the wrong text. Aware that the result could derive from a conceptual mistake, I went back to the output file, and I noticed that in all the extractions where the values were against my theory, there was also a good number of missing occurrences in the non-original text.

For example:

The extracted sentence is: “Now, good digestion wait on appetite, and health on both! ” (13 Tokens)

	Romeo	**Macbeth**
Order 0 Chains(tokens):	6.87812391255e-24	2.23400999768e-34
Missing occurrence:	3	0

Tab. 5.3 – Sentence sample

I know that the sentence is extracted from *Macbeth* because the missing occurrences are 0[6]. How is it possible then for the chain value to suggest that the sentence is more probable in *Romeo*? The answer lies in the source code. It was assumed that missing tokens values could be ignored in case of missing correspondence, considering that the numbers lack decreases with the increment of data. In this case, however, the data is not big enough even in the AC, and the number of words shared by the two texts is relevant enough to have several values ignored during multiplication.

At this point it is interesting to see the results of another analysis, in which the missing values are considered differently.

5.2 Second test and discussion

The value that most influenced the analysis seemed to be the number of missing

6 Extracting a sentence from a corpus, all the tokens in that sentence will be found in the corpus.

occurrences into our comparison; I decided at this point to assign a different value to all these occasional events. A missing occurrence can be seen as something that appears less than an hapax into a text, consequently its value has to be close with the one of an hapax. After some small computations, I considered the probability of an hapax small enough to be used for the missing occurrences as well.

The relative frequency of a word is computed dividing the number of its occurrences in the text with the number of words in the whole corpus. In this case, if the number of words of the CC is around 6,200, then the probability of a word that appears only once is 1/6,200 = 0.00016. Considering that the probability of missing words has to be smaller but still close, I decided to assign a value of 0.00015 to the missing occurrences I might find. Something similar was done for the AC, where the number of words is around 600.000. The value assigned this time, obtained in the same way as before, is then 0.000002. Another change that occurred, even if it is not considered very relevant, is the number of sentences extracted from the AC. I deemed 250 to be a very low number, considering the number of sentences in that corpora. The new number of extracted sentences is then 500.

These are the results for the characters corpora:

	Fully Random Phrases		**Controlled Length Phrases**	
From the first text:				
Order 0 Chains(tokens)	**88.75%**	11.25%	**98.75%**	8.75%
Order 0 Chains(POS)	**70%**	73.75%	**67.5%**	70%
Order 1 Chains(tokens)	**100%**	16.25%	**100%**	7.5%
Order 1 Chains(POS)	**81.25%**	63.75%	**81.25%**	56.25%
From the second text:				
Order 0 Chains(tokens)	**83.75%**	12.5%	**92.5%**	0.125%
Order 0 Chains(POS)	**52.5%**	47.5%	**67.5%**	56.25%
Order 1 Chains(tokens)	**98.75%**	18.75%	**100%**	16.25%
Order 1 Chains(POS)	**76.25%**	48.75%	**82.5%**	48.75%

Tab. 5.4 – Characters corpora second list of results. (in black new values)

And these are the results for the AC:

	Fully Random Phrases		**Controlled Length Phrases**	
From the first text:				
Order 0 Chains(tokens)	**93.4%**	49.6%	**93.6%**	50%
Order 0 Chains(POS)	**71%**	72.8%	**66.8%**	60.8%
Order 1 Chains(tokens)	**99.8%**	16.4%	**99.6%**	12%
Order 1 Chains(POS)	**85.8%**	82.8%	**81.8%**	78.4%
From the second text:				
Order 0 Chains(tokens)	**86.8%**	18.8%	**86.6%**	20.4%
Order 0 Chains(POS)	**63.8%**	61.6%	**70%**	74.4%
Order 1 Chains(tokens)	**97.6%**	15.6%	**99.4%**	18.8%
Order 1 Chains(POS)	**60.2%**	59.2%	**65.2%**	62.8%

Tab 5.5 – Authors corpora second analysis output (In black new values)

The situation is now completely changed. Looking at the CC, if before I had a good range of results - around 55% - 60% - in the POS chains, with very low values with Words, now the Words chains are really close to 100%. Also, the POS chains have increased their values, now around 75%.

Looking at the results for the AC, the increment is equally clear: in the first case, when I extracted the sentences from the Doyle's collection, the value for the order 0 word's chain was 49%, the same value calculated on sentences extracted from Stevenson was only 18%. As it so happened when compared CC, the number of missing occurrences has been significant enough to distort our results. The counter evidence to this hypothesis can be found in the results obtained from the same function as applied to the POS. Both the extractions' values are between 60% and 75% for both order 0 and 1 chains. These values are finally what I was hoping to find, even if I am aware that in this case too they have to be considered as distorted: since the number of POS is limited, and the probabilities of finding a missing occurrences are really high, we can consider this value as more reliable than any other one.

It is necessary to highlight that high values like the 1.0 in the word's chains do not have to

be considered better than the first series of values. The number of missing occurrences is too relevant in our computation to consider the words' values in both analysis as reliable. These results are not better, they are just different.

5.3 Overall discussion about pilot tests

It is interesting to see how the values are changed in the two comparisons in terms of percentage.

	Fully Random Phrases	**Controlled Length Phrases**
From the first text:		
Order 0 Chains(tokens)	688%	1000%
Order 0 Chains(POS)	5%	3%
Order 1 Chains(tokens)	513%	1200%
Order 1 Chains(POS)	27%	44%
From the second text:		
Order 0 Chains(tokens)	570%	7300%
Order 0 Chains(POS)	100%	200%
Order 1 Chains(tokens)	426%	515%
Order 1 Chains(POS)	56%	69%

Tab 5.6 – Characters corpora, percentage of increment in the second analysis

	Fully Random Phrases	**Controlled Length Phrases**
From the first text:		
Order 0 Chains(tokens)	88%	87%
Order 0 Chains(POS)	2%	9%
Order 1 Chains(tokens)	500%	730%
Order 1 Chains(POS)	3%	4%
From the second text:		

Order 0 Chains(tokens)	360%	324%
Order 0 Chains(POS)	3%	5%
Order 1 Chains(tokens)	525%	428%
Order 1 Chains(POS)	1%	3%

Tab 5.7 – Authors corpora, percentage of increment in the second analysis

These tables clearly show the differences in terms of results between tokens and POS. In the first case, they produced a top value of 7300%, totally out of control, while the results in the second case are never higher than 66%. The reason is pretty much obvious: taking into account elements like missing occurrences can completely change the results, their occurrence is almost 1 per sentence, and they make a really big difference in terms of computation. Moreover, I must notice how the differences are that high only when looking at the tokens. This category is potentially infinite, unlike the POS that are a closed group. Accordingly, Missing Occurrences are more likely in the first group.

Another thing that deserves to be mentioned is the huge difference of values between the analyses of the two pairs of corpora. In the smaller collections, values are significantly higher than what I expected, while in the AC values for the POS chains are never higher than 9%. This value strongly support the assumption I made at the beginning of this paper: it implies not only that the number of missing occurrences did not influence the computations for POS chains, but also that I can consider this theory correct, without any doubts about having directed the results after the changes we made in the second attempt. We can consider, for these reasons, the POS chains influenced by the missing occurrences only with very small collections of data.

In all the analysis, the sequence of parts of speech was associated with the right corpus at least more than 60% of the time, with a highest value of 85%. The problem of data sparseness did not seem to influence the computation, instead, looking at the last test, the sentence's length seemed to pose a more serious problem. The results I had in the first comparison are very close for both the extractions I made; the AC, instead, seem to have a step in between due to the different composition of phrases. These values could also indicate, perhaps, that the difference between sentence composition in *Macbeth* and in *Romeo and Juliet* is wider than that between Sir Arthur Conan Doyle and Robert Louis Stevenson.

6 – Results

Encouraged by the results obtained with the pilot tests, I defined all the parameters for the final analysis: the relative frequency assigned to the missing occurrences was the same as for the hapaxes. Remembering that this time the 10% of sentences has been put aside as test set, that value was computed by the code itself during the analysis.

The data, in this phase of the project, was no longer taken from the raw corpora, instead it was collected from some files in CoNLL format[7]. Because of this change, POS-tag and Syntax chains will not be computed every time, decreasing the script's execution time.
Moreover, in order to optimize the execution time, I created a Python script to put aside some elements before proceeding with the main analysis; the time spent for the computation was longer than two days for the AC too much for a common computer, so I moved some basic functions to auxiliary scripts. Specifically, I saved on a pickle file[8] the whole data structure, making it immediately available for the main script.

As last point, I would like to focus on the length of sentences. This time the analysis on CC was made on both upper and lowercased corpora, due to the small amount of data I have been able to collect. Looking at the CoNLL file, both words and sentences have suffered problems during the sentence splitting and the POS-Tagging, making sentences longer. I decided then to consider sentences with no minimum length. In addition to this, this time the test sentences are not included in the training set, so it might be possible to have missing occurrences also when comparing an element with the corpus it belongs to.

6.1 Global results

Before looking at the results, it is very important to remark that AC and CC represent two different tasks. The two CC contain collections of sentences of distinct characters, both created from the same author; what I want to see then is the level of characterization that a fictional character might have. The AC instead represent two different real persons, and the analysis aim to demonstrate in which percentage they are similar.

In other words, I can say that in the first case I want to see how different they are, in the second case how similar they are.

7 The CoNLL is a format used to annotate together text and some features. It has been developed during the The Conference on Computational Natural Language Learning in 2006, and presented one year later by (Nivré et. Al, 2006).

8 The python function cPickle – evolution of the old Pickle function – allows users to save some structured data on a file, allowing other scripts to read it. Further informations are available on Python's official web site.

6.1.1 Characters corpora results

These are the global results obtained with the CC:

Considering the Order 0 functions, the results are:

Tokens:	30/35	85%
Characters:	27/35	77%
Parts of Speech:	24/35	68%
Dep. relations:	23/35	65%
Combined classifiers:	28/35	80%
Local Voting System:	29/35	82%

Considering the Order 1 functions, the results are:

Tokens:	27/35	77%
Characters:	22/35	62%
Parts of Speech:	28/35	80%
Dep. relations:	29/35	82%
Combined classifiers:	24/35	68%
Local Voting System:	28/35	80%
Global Voting System:	27/35	77%
Draws:	5	

Tab. 6.1 – Romeo's results.

And:

Considering the Order 0 functions, the results are:

Tokens:	21/35	60%
Characters:	20/35	57%
Parts of Speech:	20/35	57%

Dep. relations:	14/35	40%
Combined classifiers:	17/35	48%
Local Voting System:	21/35	60%

Considering the Order 1 functions, the results are:

Tokens:	14/35	40%
Characters:	20/35	57%
Parts of Speech:	26/35	74%
Dep. relations:	20/35	57%
Combined classifiers:	13/35	37%
Local Voting System:	18/35	51%
Global Voting System:	18/35	51%
Draws:	4	

Tab. 6.2 – Macbeth's results.

It can be seen that the type of output has been changed. For each quintuple of classifiers the voting system returns an input, and at the end the global system accuracy is shown. The voting system is computed only on five classifiers, for instance for each iteration it can only succeed or not. In the system accuracy however, the number of classifiers is an even number, so the possibility of a draw was considered. I decided not to investigate more on draws for a specific reason: none of the classifiers, before the tests, was considered as more important than the others, and each of them returns an output that belongs to a different scale of magnitude.

Globally, these are the average results for the CC:

Considering the Order 0 functions, the results are:

Tokens:	72.5%
Characters:	67%
Parts of Speech:	62.5%

Dep. Relations:	52.5%
Combined classifiers:	64%
Local Voting System:	71%

Considering the Order 1 functions, the results are:

Tokens:	58.5%
Characters:	59.5%
Parts of Speech:	77%
Dep. Relations:	69.5%
Combined classifiers:	52.5
Local Voting System:	65.5%
Global Voting System:	64%
Draws:	4.5

Tab. 6.3 – Characters Corpora average results.

6.1.2 Authors Corpora results

The following results show the accuracy of the AC:

Tokens:	1831/2000	91%
Characters:	1566/2000	78%
Parts of Speech:	1403/2000	70%
Dep. relations:	1365/2000	68%
Combined classifiers:	1364/2000	68%
Local Voting System:	1692/2000	84%

Considering the Order 1 functions, the results are:

Tokens:	1707/2000	85%

Characters:	1686/2000	84%
Parts of Speech:	1305/2000	65%
Dep. relations:	1424/2000	71%
Combined classifiers:	1378/2000	68%
Local Voting System:	1710/2000	85%
Global Voting System:	1659/2000	82%
Draws:	148	

Tab. 6.4 – Doyle's results.

And:

Considering the Order 0 functions, the results are:

Tokens:	1735/2000	86%
Characters:	1153/2000	57%
Parts of Speech:	1025/2000	51%
Dep. relations:	1123/2000	56%
Combined classifiers:	961/2000	48%
Local Voting System:	1406/2000	70%

Considering the Order 1 functions, the results are:

Tokens:	1696/2000	84%
Characters:	1399/2000	69%
Parts of Speech:	1435/2000	71%
Dep. relations:	1201/2000	60%
Combined classifiers:	1130/2000	56%
Local Voting System:	1590/2000	79%

Global Voting System:	1423/2000	71%
Draws:	183	

Tab. 6.5 – Stevenson's results.

Making the number of sentences extracted explicit, the difference between the two analysis, in terms of quantity, should be clear.

Again, is interesting to see the average results:

Tokens:	88.5%
Characters:	67.5%
Parts of Speech:	60.5%
Dep. relations:	61%
Combined classifiers:	58%
Local Voting System:	77%

Considering the Order 1 functions, the results are:

Tokens:	84.5%
Characters:	76.5%
Parts of Speech:	68.5%
Dep. Relations:	65.5%
Combined classifiers:	62%
Local Voting System:	82%
Global Voting System:	76.5%
Draws:	165.5

Tab. 6.6 – Authors Corpora average results.

6.2 Discussion

The results of the second analysis are perfectly in line with the expectations. The CC show an inconsistent behaviour, due to the lack of data they suffered. Considering the single classifiers, some assumptions can be made. First of all, tokens are strongly relevant on order 0 chains, and quite relevant on order 1 chains. This indicates a strong differentiation of vocabulary between the corpora, even if they use the same English. The different context in which Romeo and Macbeth are inserted are, very likely, a determining factor of distinction. On the contrary, characters are in three out of four cases, representative of the other classifiers average. On the other side, parts of speech and syntax elements confirm their high grade of success. These values are particularly relevant because they are not influenced by missing occurrences; both POS and Syntactic elements chains are composed by closed groups of elements, accordingly the missing occurrences are very rare events and they do not influence the analysis result.

Very surprisingly, looking at the combination of classifiers value, the results are completely discording. In Romeo's analysis, I have an 80% of right assignments for the order 0 chains and 68%

for the order 1; in Macbeth, however, order 0 chains have given 48% of right assignments, and order 1 chains have given 37%. There is a difference of 31 percent points in a case and even 40 percentage points in another. The only way to consider these informations as reliable is to justify this difference; this has been done taking into consideration only the sentences extracted from Macbeth. The sentences composing the Test Set were really short, as for instance “Filthy hags!”, “What is't you do?”, “What is the night?” or “Unreal mockery, hence!”. Short sentences are in the majority of cases not relevant from a linguistic point of view; evidently in this case the random extraction was not particularly lucky, and led to these results.

Going back to the results, the voting system still needs to be considered. Again, as I have shown above, the results in Romeo are significantly higher than in Macbeth. The important thing to notice is that the value of the voting system in almost all cases is higher than the average of all the five classifiers it was related to.

Something surprising is that in each case, the success rate of the voting system is always higher than the corresponding combination of classifiers. The combination of classifiers is computed multiplying the value of the other four classifiers, so it should increase the difference between the higher and the lower value. The voting system instead, indicates how many classifiers have given the right response, no matter how big the difference between the classifier was.

Even considering this a good result, on the other hand the result of the global system accuracy cannot be ignored. In this case the result of 51 percent indicates almost a random assignment. The combination of all the classifiers in this case has proven to be completely unreliable.

Looking at the AC the situation completely changes. Also in this case the analysis has shown two different results; this time, however, both analyses have given encouraging results. Tokens and Characters classifiers have provided very good results, with only one exception. On average, the percentage of success of these classifiers is 79.25%, which indicates a strong relevance in the identification of the author. The POS analysis has given a good output only when tested on order 1 chains, with a 65 and a 71% of right assignments. Results concerning dependency relations and the combination of classifiers instead reveal a different trend: in Doyle's corpus they have been quite satisfying, even if the results are not as high as I expected; in Stevenson's corpus instead values are between 48 and 60 percent, surely not as relevant as they should have been.

As in the case of CC, the voting system has returned results that are higher than the average of other classifiers combinations. Results are always between 70 and 85 percent, with only one value lower than 79 percent, undoubtedly a great result. The system accuracy, with percentages of 71 and 82, completes this promising scenario.

Globally, both analysis have returned interesting results. Some classifiers, especially Tokens and both voting systems, have demonstrated a high reliability, regardless the type of corpora taken into account. All these results are encouraging because the two pairs of corpora are representative of two different tasks, based on different types of material (the order of magnitude of CC and AC are significantly different). The similarity between the results then, implies that the system is elastic enough to be functional with different types of material.

7 – Conclusions and further developments

Comparing the data gained with the analysis, the global result can be considered quite satisfying. The accuracy, in the majority of cases, is satisfying, and the output is in line with what I expected to obtain. The lack of data in Shakespeare's corpora, together with the type of English used, significantly affected some classifiers. However, looking at the AC, it is clear that with a proportionate amount of data, this type of quantitative analysis can be a good starting point to identify the style of a writer or a character.

This work, in conclusion, shows how some constituents can be relevant to identify the style of an author or how they can at least be indicative of him and his style. Nonetheless, the role played by semantics in all those theories must not be overlooked. As stated in the introduction, it is clear that semantics and phrasal structure must be considered as equal members of the same system. My purpose was to understand in which manner these classifiers work in a “stand alone” context, with some light attempts to mix them, and the results I found have revealed that the order of all these elements are, to identify a precise writing style, only partly to be trusted. There are still some important elements to consider in future analysis, as the role of punctuation of the dynamics due to the number of shared words in the Vocabulary, but this can still be considered as a good starting point. The missing occurrences, instead, might be analysed not as mere concatenation of characters with a missing occurrence into a corpus, but more as elements with a semantic meaning, with synonyms and antonyms, as proposed by Li. Tools like Google books could be used in future, together with structured lexical databases – like WordNet – to reduce the effect of these missing elements, making a program able to substitute a word that he does not know, with a similar one, without affecting the result of the analysis.

Appendix I – scripts list

The list shown below summarize the function of some script made by the author.

MacbethScraper.py, *HamletScraper.py*, *RomeoScraper.py* : All these files have the same structure, one main function that, starting from a given URL, grab all the nodes containing the sentences attributed to a character. Once all the sentences have been grabbed, they create a file containing all those sentences.

Pickler.py: This function takes a CoNLL file as input, and returns a .p file containing the whole data in "pickle" format, making the execution of the main script faster.

PosTagger.py: This script has the same purpose of *Pickler.py*, taking as input a CoNLL file, it returns a .p file containing an array of Part of Speech.

TextCleaner.py: This small script has been used to normalize the corpora, delete some annotations, and convert non UTF-8 characters.

Dissertation.py: This is the main file, the one that makes the biggest work. It takes two CoNLL files as input, and using the Class *Corpus* and its methods, extracts the first set of features. The function *analyser* takes as input the two Classes previously created and run the real analysis, dividing training set and test set, and using Markov-Chains based Classifiers of order 0 and order 1, it returns some variables representing the score of the analysis. The function *printingmess* print on the screen the whole output in a pleasant format.

Appendix II – Corpus of Characters Results

Statistical analysis on Romeo and Macbeth

	Romeo	**Macbeth**
Number of sentences:	297	310
Number of tokens:	5126	6009
Number of characters:	17760	21223
Number of hapax:	871	1096
Hapaxes Distribution:	0.169918064768	0.182393077051
Sentences length average:	72.7609427609	83.2161290323
Tokens length average:	3.46468981662	3.53186886337
Vocabulary's size:	1362	1637
Grammatical Tokens Perc.:	45.8137510879	46.8764460898
Type Token Ratio:	0.265704252829	0.272424696289
Lexical Density:	0.609535304768	0.582336382829

Deeper analysis on Romeo's corpus:

Considering only Nouns and Adjectives,
the first ten bigrams in order of frequence are:

1)	('NN', 'NN')	472
2)	('JJ', 'NN')	213
3)	('NN', 'NNP')	194
4)	('NN', 'JJ')	191
5)	('NNP', 'NN')	186
6)	('NN', 'NNS')	91

7)	('NNS', 'NN')	72
8)	('NNP', 'JJ')	67
9)	('JJ', 'JJ')	52
10)	('NNP', 'NNP')	52

The first ten named people are:

1)	Tybalt	7
2)	Farewell	6
3)	Romeo	5
4)	Good	3
5)	Juliet	3
6)	Rosaline	3
7)	Thou	3
8)	Dost	2
9)	Hath	2
10)	Love	2

The first ten named places are:

1)	Juliet	11
2)	Than	3
3)	Verona	3
4)	Mercutio	2
5)	More	2
6)	Nor	2
7)	Tybalt	2
8)	Being	1
9)	Benvolio	1
10)	Displant	1

On 147 occurrences, "I" is the subject 140 times,
corresponding to the 95 percent.

Globally, on 545 subjects, "I" is the subject 147 times,
corresponding to the 25 percent.

On 545 detected subjects, the average distance from the sentence's head is 2.48073394495.

1) NN - 914
2) IN - 469
3) , - 443
4) PRP - 414
5) DT - 396
6) JJ - 336
7) . - 327
8) RB - 289
9) VB - 286
10) CC - 193
11) NNP - 180
12) NNS - 178
13) : - 173
14) VBP - 162
15) PRP$ - 160
16) VBZ - 137
17) MD - 103
18) VBD - 95
19) TO - 94
20) VBN - 71

Deeper analysis on Macbeth's corpus

Considering only Nouns and Adjectives,
the first ten bigrams in order of frequence are:

1) ('NN', 'NN') 483

2)	('JJ', 'NN')	224
3)	('NN', 'NNP')	209
4)	('NNP', 'NN')	194
5)	('NN', 'JJ')	193
6)	('NN', 'NNS')	102
7)	('NNS', 'NN')	89
8)	('NNP', 'JJ')	72
9)	('NNP', 'NNP')	71
10)	('JJ', 'JJ')	63

The first ten named people are:

1)	Thou	7
2)	Till	6
3)	Banquo	5
4)	Seyton	4
5)	Which	4
6)	Will	4
7)	Macbeth	3
8)	Macduff	3
9)	Shall	3
10)	Thy	3

The first ten named places are:

1)	Cawdor	7
2)	Banquo	5
3)	Duncan	4
4)	Whose	4
5)	Birnam	3
6)	Dunsinane	3
7)	Thou	3
8)	England	2
9)	Fleance	2

10) That 2

On 144 occurrences, "I" is the subject 134 times,
corresponding to the 93 percent.

Globally, on 641 subjects, "I" is the subject 144 times,
corresponding to the 20 percent.

On 641 detected subjects, the average distance from the sentence's head is 2.01560062402.

1) NN - 921
2) IN - 510
3) DT - 503
4) PRP - 475
5) , - 457
6) JJ - 372
7) . - 344
8) VB - 341
9) RB - 291
10) NNP - 230
11) CC - 228
12) NNS - 224
13) : - 205
14) VBP - 203
15) PRP$ - 189
16) VBZ - 151
17) MD - 146
18) TO - 124
19) VBN - 111
20) VBD - 96

```
#####################################
#                                   #
#              Final Results        #
#                                   #
#####################################
```

Romeo's results:

Considering the Order 0 functions, the results are:

Tokens:	30/35	85%
Characters:	27/35	77%
Parts of Speech:	24/35	68%
Dep. relations:	23/35	65%
Combined classifiers:	28/35	80%
Local Voting System:	29/35	82%

Considering the Order 1 functions, the results are:

Tokens:	27/35	77%
Characters:	22/35	62%
Parts of Speech:	28/35	80%
Dep. relations:	29/35	82%
Combined classifiers:	24/35	68%
Local Voting System:	28/35	80%
Global Voting System:	27/35	77%
Draws:	5	

Macbeth's results:

Considering the Order 0 functions, the results are:

Tokens:	21/35	60%
Characters:	20/35	57%
Parts of Speech:	20/35	57%
Dep. relations:	14/35	40%
Combined classifiers:	17/35	48%
Local Voting System:	21/35	60%

Considering the Order 1 functions, the results are:

Tokens:	14/35	40%
Characters:	20/35	57%
Parts of Speech:	26/35	74%
Dep. relations:	20/35	57%
Combined classifiers:	13/35	37%
Local Voting System:	18/35	51%
Global Voting System:	18/35	51%
Draws:	4	

Work started at 2014-01-16 10:25:46.087950.

Work ended at 2014-01-16 10:26:19.260874.

Appendix III – Corpus of Authors Results

Statistical analysis on Sir Arthur Conan Doyle and Robert Louis Stevenson

	Doyle	**Stevenson**
Number of sentences:	23762	18634
Number of tokens:	569281	544221
Number of characters:	2102765	2012940
Number of hapax:	10715	14152
Hapaxes Distribution:	0.0188219877354	0.0260041416998
Sentences length average:	108.648598603	131.927659118
Tokens length average:	3.69372067573	3.69875473383
Vocabulary's size:	22902	28237
Grammatical Tokens Perc.:	48.2683585834	50.3061603709
Type Token Ratio:	0.0402296932446	0.051885171649
Lexical Density:	0.545024788949	0.530360546563

Deeper analysis on Doyle's corpus

Considering only Nouns and Adjectives,
the first ten bigrams in order of frequence are:

1)	('NN', 'NN')	51310
2)	('JJ', 'NN')	18775
3)	('NN', 'JJ')	17336
4)	('NN', 'NNP')	11750
5)	('NNP', 'NN')	10296
6)	('NNP', 'NNP')	9748
7)	('NNS', 'NN')	8235
8)	('NN', 'NNS')	8102
9)	('JJ', 'JJ')	4640
10)	('JJ', 'NNS')	4058

The first ten named people are:

1)	Holmes	1351
2)	Watson	657
3)	Mr. Holmes	311
4)	Sherlock Holmes	164
5)	Lestrade	157
6)	Sir Henry	124
7)	Sir Charles	74
8)	Stapleton	72
9)	Mr. Sherlock Holmes	57
10)	Baker Street	47

The first ten named places are:

1)	London	274
2)	Holmes	114
3)	England	99
4)	English	50
5)	America	41
6)	American	38
7)	Indian	36
8)	French	34
9)	British	32
10)	Chicago	32

On 13328 occurrences, "I" is the subject 12981 times,
corresponding to the 97 percent.

Globally, on 65975 subjects, "I" is the subject 13328 times,
corresponding to the 19 percent.

On 65975 detected subjects, the average distance from the sentence's head is 1.68410761652.

The first twenty POS, ordered by frequency, are:

1) NN - 71526
2) IN - 64947
3) DT - 55023
4) PRP - 53201
5) , - 35734
6) VBD - 35305
7) . - 35022
8) JJ - 32003
9) RB - 29884
10) NNP - 21957
11) CD - 21095
12) VB - 20230
13) CC - 18613
14) PRP$ - 15802
15) NNS - 15485
16) VBN - 14464
17) TO - 12488
18) VBP - 11478
19) MD - 9945
20) VBZ - 9481

Deeper analysis on Stevenson's corpus

Considering only Nouns and Adjectives,
the first ten bigrams in order of frequence are:

1)	('NN', 'NN')	48082
2)	('NNP', 'NNP')	18723
3)	('JJ', 'NN')	17910

4)	('NN', 'JJ')	16503
5)	('NN', 'NNP')	11107
6)	('NNP', 'NN')	10067
7)	('NN', 'NNS')	9559
8)	('NNS', 'NN')	9224
9)	('JJ', 'NNS')	4319
10)	('JJ', 'JJ')	4260

The first ten named people are:

1)	Dick	777
2)	Jim	241
3)	Sir Daniel	213
4)	Mr. Henry	168
5)	Carthew	167
6)	Pinkerton	164
7)	Silver	142
8)	Loudon	112
9)	Matcham	103
10)	Nares	95

The first ten named places are:

1)	French	96
2)	English	69
3)	Paris	59
4)	England	52
5)	American	49
6)	European	44
7)	Bellairs	37
8)	Holywood	37
9)	Durrisdeer	36
10)	Indian	34

On 11424 occurrences, "I" is the subject 11053 times,
corresponding to the 96 percent.

Globally, on 56806 subjects, "I" is the subject 11424 times,
corresponding to the 19 percent.

On 56806 detected subjects, the average distance from the sentence's head is 2.13514417491.
The first twenty POS, ordered by frequency, are:

1) NN - 75411
2) IN - 62487
3) DT - 58233
4) , - 42768
5) PRP - 42172
6) VBD - 35806
7) JJ - 31605
8) RB - 29172
9) . - 25123
10) CC - 24492
11) NNP - 20590
12) VB - 17953
13) NNS - 17700
14) PRP$ - 13786
15) VBN - 13210
16) " - 11693
17) TO - 11643
18) : - 10655
19) CD - 9319
20) VBP - 8925

```
#####################################
#                                   #
#              Final Results        #
#                                   #
#####################################
```

Doyle's results:

Tokens:	1831/2000	91%
Characters:	1566/2000	78%
Parts of Speech:	1403/2000	70%
Dep. relations:	1365/2000	68%
Combined classifiers:	1364/2000	68%
Local Voting System:	1692/2000	84%

Considering the Order 1 functions, the results are:

Tokens:	1707/2000	85%
Characters:	1686/2000	84%
Parts of Speech:	1305/2000	65%
Dep. relations:	1424/2000	71%
Combined classifiers:	1378/2000	68%
Local Voting System:	1710/2000	85%
Global Voting System:	1659/2000	82%
Draws:	148	

Stevenson's results:

Considering the Order 0 functions, the results are:

Tokens:	1735/2000	86%
Characters:	1153/2000	57%
Parts of Speech:	1025/2000	51%
Dep. relations:	1123/2000	56%
Combined classifiers:	961/2000	48%
Local Voting System:	1406/2000	70%

Considering the Order 1 functions, the results are:

Tokens:	1696/2000	84%
Characters:	1399/2000	69%
Parts of Speech:	1435/2000	71%
Dep. relations:	1201/2000	60%
Combined classifiers:	1130/2000	56%
Local Voting System:	1590/2000	79%
Global Voting System:	1423/2000	71%
Draws:	183	

Work started at 2013-12-22 10:29:07.800135.

Work ended at 2013-12-24 04:41:18.947753.

Bibliography

Argamon, S., Levitan, S. 2005. Measuring the usefulness of function words for authorship attribution, *Proceedings of the ACH/ALLC Conference*, Victoria, BC, Canada.

Argamon, S., Koppel, M., Pennebaker, J. and Schler, J. (2008), Automatically Profiling the Author of an Anonymous Text, *Communications of the ACM*, 52 (2):119–123, 2009

Attardi G. 2006. Experiments with a multi-language non-projective dependency parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X '06)*, NewYork City, New York, 166–170.

Attardi G., Dell'Orletta F., Simi M., Turian J.. 2009. Accurate Dependency Parsing with a Stacked Multilayer Perceptron. In: Proceedings of Evalita'09

Bar-Hillel, Y. 1954. *Logical Syntax and Semantics*. Language. Vol. 30 No. 2 (Apr. - Jun.), Washington: linguistic society of America

Bethard, S. , Yu, H. , Thornton, A. , Hatzivassiloglou, V. and Jurafsky, D., 2004. Automatic extraction of opinion propositions and their holders. In AAAI Spring Symposium on Exploring Attitude and Affect in Text

Bhosale, Shruti, Heath Vinicombe & Raymond Mooney (2013). Detecting promotional content in Wikipedia. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Wash., 18–21 October 2013, pp. 1851–1857

Brooke J., Hammond A. and Hirst G., 2013, clustering voices in the waste land, in *Proceedings of the 2nd Workshop on Computational Literature for Literature* (CLFL '12), Montreal

Chaski, C. 2005. Who's at the keyboard? Authorship attribution in digital evidence investigations. I*nternational Journal of Digital Evidence*, 4(1)

Chomsky, N., 1957, *Syntactic Structures*, Berlin: Mouton de Gruyter.

Church, K. and Mercer, R., 1993, Introduction to the Special Issue on Computational Linguistics Using Large Corpora, Massachusetts: MIT Press Cambridge

Congzhou He Ramyaa and Khaled Rasheed. 2004. Using machine learning techniques for stylometry. In *Proceedings of International Conference on Machine Learning*, (MLMTA'2004). Las Vegas.

Cox, D. R. and Brandwood, L. (1959). On a Discriminating Problem Connected with the Works of Plato. Journal of the Royal Statistical Society B, 21:195-200

Davies, M., 2009, The 385+ million word *Corpus of Contemporary American English* (1990-2008+): Design, architecture, and linguistic insights, *International Journal of Corpus Linguistics*, Vol. 14, Number 2, Philadelphia: John Benjamins Publishing Company.

de Morgan, S. E. (1882). Memoir of Augustus de Morgan by his Wife Sophia Elizabeth de Morgan With Selections From His Letters. Longmans, Green, and Co., London.

Dell'Orletta F.. 2009. Ensemble system for Part of-Speech tagging. In *Proceedings of Evalita'09, Evaluation of NLP and Speech Tools for Italian* Reggio Emilia, December.

Elson D. and McKeown K., 2010. Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth* AAAI Conference on Artificial Intelligence (AAAI 2010),Atlanta, Georgia.

Francis, W. N. and Kucera, N., 1979, *Brown Corpus Manual*, Providence, Rhode island: Brown University

Firth, J.R., 1957, A synopsis of Linguistic Theories, *Studies* in *Linguistic Analysis*, Oxford: Philological Society

Gildea, Daniel. 2001. Corpus variation and parser performance. In *Proceedings of 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Pittsburgh, PA

He H., Barbosa D., and Kondrak G.. 2013. Identification of speakers in novels.The 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)

Hepple, M., 2000, Independence and commitment: assumptions for rapid training and execution of rule-based POS-taggers, *ACL '00 Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, Stroundsburg: Association for Computational Linguistics.

Holmes, D. I., 1994, Authorship Attribution, *Computers and the Humanities*, Vol. 28, No. 2, New York: Springler.

Holmes, D.I., 1998. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, Oxford: Oxford Journal

Juola, P., 1998. Cross-entropy and linguistic typology, *Proceedings of New Methods in Language Processing* 3. Sydney, Australia

Juola, P. and Baayen, H., 2005, A Controlled-corpus Experiment in Authorship Identification by Cross-Entropy, L*iterary and Linguistic Computing*

Khmelev D.V., 2001, Disputed Authorship Resolution through Using Relative Empirical Entropy for Markov Chains

of Letters in Human Language Text, J*ournal of Quantitative Linguistics*, 7(3)

Koppel, M., Argamon, S., & Shimoni, A. (2002). Automatically categorizing written texts by author gender. Literary and Linguistic Computing, 17(4)

Lebert, M., 2008, *Project Gutenberg (1971-2008),* eBook version from enebooks.com

Li, W., 1992, Random texts exhibit Zipf's-law-like word frequency distribution, *,IEEEE transaction on information theory,* Vol. 38 N. 6

Li, et al. August 2006, Sentence Similarity Based on Semantic Nets and Corpus Statistics *,IEEEE transaction on knowledge and data engineering,* Vol. XVIII, N. 8, 2006

Lytinen, S.L., 1986, *Dinamically combining syntax and smenatics in Natural Language processing*, Yale University:AAAI Library

Lytinen, S.L., 1996, *Semantics-first Natural Language Processing*, Yale University:AAAI Library

Manoj Harpalani and Michael Hart and Sandesh Singh and Yejin Choi and Rob Johnson, Language of Vandalism: Improving Wikipedia Vandalism Detection via Stylometric Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (to appear), Portland, Oregon, 2011

Mendenhall, T.C., 1887, The characteristic curves of composition, *Science*, Vol. IX, 11: 237-49.

Mosteller, F. and Wallace, D. L. (1964). Applied Bayesian and Classical Inference: The Case of the Federalist Papers. Addison-Wesley, Reading, MA.

Mosteller,F . and Wallace, D. L. 1984, *Applied Bayesian and Classical Inference: The Case of the Federalist Papers*. Reading: Addison-Wesley

Nivre, J. Hall J., Kubler S., McDonald R., Nilsson J., Riedel S., and Yuret D., 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of EMNLP-CoNLL 2007*, pages 915–932.

Owens Jr, R. E. , 1984 , *Language development, an introduction,* Columbus: Merrill Pub Co

Stamatatos, E., 2009, A Survey of Modern Authorship Attribution Methods, *Journal of the American Society for Information Science and Technology*,

Thatcher, J.W. , 1967, *Characterizing Derivation Trees of Context-Free Grammars through a Generalization of Finite Automata Theory* , New York: IBM Watson Research Center.

Van Halteren, H. (2004) Linguistic profiling for authorship recognition and verification, *Proc. of 42nd Conf*. Of ACL, July 2004, pp. 199-206

Williams, C. B. (1940). A Note on the Statistical Analysis of Sentence-length as a Criterion of Literary Style. *Biometrika*, 31: 356-61.

Yule, G. U. (1938). On sentence-length as a Statistical Characteristic of Style in Prose, with Application to Two Cases of Disputed Authorship, *Biometrika*, 30: 363-90.