AUTOMATIC STYLISTIC PROCESSING FOR CLASSIFICATION AND TRANSFORMATION OF NATURAL LANGUAGE TEXT

A thesis proposal

by

Foaad Khosmood

We propose to build a modular, extensible, web-accessible software system capable of using heterogeneous methods to accomplish automatic language style-based classification and transformation.

UCSC department of Computer Science

Fall 2008
# TABLE OF CONTENTS

## 1 BACKGROUND

1.1 INTRODUCTION AND MOTIVATION 4
1.1.1 OVERVIEW OF THIS DOCUMENT 4
1.1.2 STYLE 4
1.1.3 DETECTING STYLE 5
1.1.4 STYLE-BASED TRANSFORMATION 5
1.1.5 CLASSIFICATION-TRANSFORMATION LOOP 6
1.1.6 EXTENSIBLE, MODULAR, HETEROGENEOUS SYSTEM 7

## 1.2 APPLICATIONS 8

1.2.1 DIGITAL FORENSICS AND PLAGIARISM DETECTION 8
1.2.2 STYLE OBfuscATION AND STYLE ENCRYPTION 8
1.2.3 STYLE BASED SEARCH 8
1.2.4 EDUCATIONAL AND PRODUCTIVITY TOOLS 8
1.2.5 AUTHORING TOOLS AND DIGITAL ENTERTAINMENT APPLICATIONS 9
1.2.6 MACHINE TRANSLATION ENHANCEMENT 9
1.2.7 INTERACTIVE AGENTS 9

## 1.3 RELATED WORK BY OTHERS 10

1.3.1 TEXT CLASSIFICATIONS: GENRE, TEXT TYPE AND REGISTER 10
1.3.2 STYLE AND STYLISTICS 11
1.3.3 STYLISTICS IN HUMANITIES 12
1.3.4 REACTIONS TO STYLISTIC ORIENTED METHODS 13
1.3.5 DEFENSE OF STYLISTICS 15
1.3.6 STYLE AS “OPTION” 17
1.3.7 STYLE MARKERS AND COMPUTATIONAL STYLISTICS 18

## 1.4 RELATED WORK BY US 20

1.4.1 RESEARCH IN SOURCE ATTRIBUTION (CLASSIFICATION) 20
1.4.2 RESEARCH IN STYLE PROCESSING (CLASSIFICATION AND TRANSFORMATION) 20

## 2 SYSTEM, TOOLS AND EXAMPLES 23

### 2.1 STYLISTIC CLASSIFICATION-TRANSFORMATION SYSTEM 23

2.1.1 GOALS 23
2.1.2 REQUIREMENTS 24
2.1.3 DESIGN 25
2.1.4 TRANSFORMATION OPERATORS AND TRANSFORMS 26
2.1.4.1 String replacement operators 26
2.1.4.2 Semantic operators 27

### 2.2 SAMPLE WALKTHROUGHS 28

2.2.1 MODERNIZING SHAKESPEARE 28
2.2.2 A TRIVIAL 2-MARKER SCENARIO IN DEPTH 30

### 2.3 TOOLS AND RESOURCES 32

2.3.1 AAAC CORPUS AND JGAAP 32
2.3.2 LINK GRAMMAR PARSER 33
2.3.3 PARAPHRASING AND SUMMARIZATION TOOLS 33
2.3.4 WORD NET, ONTOLOGY AND WORD LISTS 33
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.5</td>
<td>GNU DICTION AND STYLE</td>
<td>34</td>
</tr>
<tr>
<td>2.3.6</td>
<td>READABILITY MEASURES AND AUTOMATIC SCORERS</td>
<td>34</td>
</tr>
<tr>
<td>2.3.7</td>
<td>LANCASTER UNIVERSITY ONLINE COURSE ON STYLISTICS</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>CURRENT STATUS, WORK PLAN AND EVALUATION</td>
<td>35</td>
</tr>
<tr>
<td>3.1</td>
<td>CURRENT STATE OF SOFTWARE</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>PROJECT PLAN AND TIMELINE</td>
<td>37</td>
</tr>
<tr>
<td>3.3</td>
<td>EVALUATION</td>
<td>39</td>
</tr>
<tr>
<td>3.3.1</td>
<td>CLASSIFICATION</td>
<td>39</td>
</tr>
<tr>
<td>3.3.2</td>
<td>MACHINE AND HUMAN VERIFICATION FOR TRANSFORMATION</td>
<td>39</td>
</tr>
<tr>
<td>3.3.2.1</td>
<td>Machine verification of transformation</td>
<td>39</td>
</tr>
<tr>
<td>3.3.2.2</td>
<td>Human evaluation</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>SUMMARY</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>THE PROPOSAL</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>BACKGROUND</td>
<td>41</td>
</tr>
<tr>
<td>4.3</td>
<td>APPROACH</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>PLAN</td>
<td>41</td>
</tr>
<tr>
<td>4.5</td>
<td>TARGETED CONTRIBUTION</td>
<td>41</td>
</tr>
<tr>
<td>5</td>
<td>GLOSSARY</td>
<td>43</td>
</tr>
<tr>
<td>6</td>
<td>BIBLIOGRAPHY</td>
<td>44</td>
</tr>
<tr>
<td>7</td>
<td>APPENDICES</td>
<td>47</td>
</tr>
<tr>
<td>7.1</td>
<td>GETMARKERS.BASH SCRIPT</td>
<td>47</td>
</tr>
<tr>
<td>7.2</td>
<td>SAMPLE OUTPUT FROM A DEMONSTRATION PROGRAM</td>
<td>49</td>
</tr>
</tbody>
</table>
1 Background

1.1 Introduction and motivation

Abstract

Style is an integral part of natural language in written, spoken, or machine generated forms. Natural language styles are understood, mimicked and transformed by human agents with ease. We believe that like natural language processing (NLP) in general, natural language styles can also be processed, recognized, generated and transformed computationally. In this document, we propose to build a modular, extensible system that automatically performs style-based classification and transformation on written language. We call the system modular because we would like to accommodate an open set of stylistic markers, language operators and evaluation methods and reasoning methods. The system will be able to make use of heterogeneous processes to work together in a large variety of combinations in order to deliver the best possible results for style classification and transformation tasks.

1.1.1 Overview of this document

This thesis proposal is divided into three main sections: Background, System and Plan. The first is background where introduction, purpose and previous research are presented. We begin with the concept of style and build up to our proposal. When discussing relevant works, we divide the research into a survey of related literature in various fields (1.3) and research contributions done by ourselves thus far (1.4).

In the second section, we present requirements and design for the system we plan to build, including some use cases and walkthroughs. The third section is about our future roadmap, tools, timeline and evaluation plan.

The three main sections are followed by glossary, bibliography and appendices.

1.1.2 Style

Broadly speaking, style is a way of doing something. By implication that same thing can be done in more ways than one. Style is often, although not necessarily, associated with an actor, person, role or entity. For example, in the game of chess, an opening refers to the first few moves of a player. If we define chess style as having a distinct opening, we could identify the opening of a particular player’s chess style just by examining the record of a series of games by the player whom we already associate with having that style. The opening would be one recognizable factor in the “style of chess play” being observed.

All natural languages and even most artificial languages can be associated with one or more “styles.” There is no universally acknowledged definition of what specific elements
constitute “style” when it comes to language. We simply distinguish between different stylistic elements by observing the unique way in which one literary act is done, all else being equal. The notion is inherently imprecise rendering any decent contextual definition necessarily broad. Linguists often use terms like “dialect” and “register,” as concepts that fit inside the larger notion of style. Dialects are socio-geographical and strongly associated with actual agents. Register is a linguistics concept describing language along the axes of field (subject matter), tenor (formality and social relationships) and mode (medium of communication). Dialect and register are examples of concepts that fit within the larger category of style.

On the other hand, completely non-linguistic and typographical choices in written language are often called style as well. Examples include font, size, color, text decorations, visual emphasis, indentation and usage of non-linguistic symbols.

In the “background” section below, we discuss how we adapt and appropriate a suitable definition of style for this work.

1.1.3 Detecting style

Given that styles are omnipresent in language and they can be easily detected by humans, we should be able to at least partially define a series of features detectable in each distinct style. These are some of the same features that convince a human examiner to assign a distinct style label to a piece of text, considered either on its own or in comparison to one or more other texts. Fortunately, there is a large body of research in artificial intelligence, machine learning, natural language processing and computational stylistics that has considered this type of algorithmic feature selection and text classification in depth. In addition, works in linguistics, sociolinguistics, corpus linguistics, literary studies and language studies provide a rich repository of analysis about language variation that can be used to augment aforementioned algorithmic and mathematical methods. We hope to employ this literature to derive ever-richer and more complex style markers that could in turn deliver more precise classifications.

1.1.4 Style-based transformation

Style-based transformation is the idea of linguistically altering a piece of text so it exhibits the characteristics of other pieces of text which are associated with a style. It is essentially a text-to-text language translation operation similar to machine translation (MT) except that instead of translating from one language to another, it translates from one style to another. As with MT, the meaning of the original message has to be preserved as much as possible.

Unlike the classification problem, very little automatic or computational research has been attempted in this area. Overall, the most likely fields one can expect to encounter this particular exercise are in literary education and writer development. And even in those areas it is a task for human agents, not artificial ones.
However, some loosely related concepts and technologies exist in artificial intelligence and natural language processing which will be helpful to us. Natural Language Generation (NLG) can perhaps be considered “half” the problem. Once the intention or the intended meaning of the utterance is known, NLG techniques can produce multiple stylistic variants of the corresponding language. Some productivity tools and computational writing analyzers are often implemented as rule-driven expert systems capable of suggesting rewrites and paraphrases in order to improve the quality of writing in some way. Although, the goal of these rewriting routines would not be the same as what we have in mind, they nevertheless do represent automatic language alteration tools.

1.1.5 Classification-transformation loop

We contend that transformation is intimately related to classification and vice versa. Successful transformation depends on precise classification in the sense that a given classifier ultimately determines whether or not the transformed text exhibits enough markers to be considered as one of the texts of the target style. Style markers that help the classification of text, in turn, provide the basis for style transforms. This conception allows for an elegant symbiotic growth in sophistication for both sub-systems. For the design and prototyping of our system, we can use this interdependence to focus our efforts where they would produce the best results. The interdependence between classification and transformation can be illustrated by the following example:

We imagine a style we could call “Shakespearian,” derived from a corpus consisting of several of Shakespeare’s tragedies. We find that our classifier can easily distinguish between Shakespearian corpus and modern English corpus just by detecting the presence of early modern English pronoun forms (such as “thou”, “ye”, or “thine”). This gives us a clear path to build a transform which simply converts all the modern pronouns in a given text to the old English equivalent. The transformed text would now be classified in our own system as “Shakespearian.” However, when read by human experts, the text would be far from Shakespearian; after all, there is more to Shakespeare than pronouns.

We conclude that the classification is in fact incorrect because the style of Shakespearian Tragedy was grossly underspecified. The corrective action, then, would be to add more style markers that specifically discriminate between the once-transformed text and the Shakespearian corpus. We can do this by observing more unique features of Shakespeare. Examples of these features are: iambic pentameter, presence of dramatic characters, playwright-style text divisions (i.e. acts and scenes), distinct vocabulary, motifs of violence, death or betrayal, etc. Each of these observations provides two things for us simultaneously:

1. The basis for a marker that can be analyzed by our classifier
2. The basis for a transform that is capable of altering text such that the result exhibits the phenomenon in question

The second task requires progressively sophisticated linguistic text-to-text operators. To be sure, not all style markers could yield related operators without those operators also
producing significant side-effects. Changing general modern prose to iambic pentameter would be extremely difficult if not impossible. But the point is that each marker embeds within it a related transform. And each transformation process produces text that might still be lacking the desired style as observed by a human expert, prompting for more complete style specification through finding better markers. This is a spiral that can continue as long as there are creative ways to make markers and transforms, or until such time as a reasonable human observer could confidently decide that a transformed piece of text is “Shakespearian enough.”

1.1.6 Extensible, modular, heterogeneous system

We are interested in building a system capable of performing stylistic analysis, classification and classification-aided transformation as described above. But we also want the system to have maximum flexibility to analyze a large set of markers (or intelligently chosen subsets) and to employ a large number of transforms. We would like the system to take advantage of markers and transforms that have been developed from a wide variety of sources for different purposes and apply them to tasks as required. Lastly, we would like the system to be easily augmentable. We envision a web accessible knowledge based system where anybody can submit new markers and transforms and watch the system perform.

The above requirements necessitate the use of well defined layers of abstraction. A standard interface for style markers and style transforms is necessary in order to exploit heterogeneous operations. In addition to markers and transforms, the classification, transformation and evaluation methods themselves should be extensible. The system should allow plugging in off-the-shelf machine learning and clustering algorithms for its classification section. Similarly, optimizers and planning agents could be enhanced to make transformation more efficient.
1.2 Applications

Abstract
Some applications that we foresee being able to make use of our system include digital forensics and plagiarism, authorial style obfuscation (for privacy purposes), style based search, authoring tools for development of narratives and personalities in writing and digital entertainment, machine translation, writing/educational tools and adaptable computer agents for human interaction.

1.2.1 Digital forensics and plagiarism detection

The classifier part of the system could be used to identify and match the style of a text to one of the styles derived from examining corpora belonging to two or more suspect authors. Similarly in plagiarism detection, the system could match the style of the author being plagiarized against the plagiarized text and even provide the style markers that can serve as evidence to show plagiarism.

1.2.2 Style obfuscation and style encryption

For a variety of reasons people may wish to intentionally obfuscate the style of their own writing. Style remains an easy way to recognize the originator of an anonymous piece of writing by humans and machines. Our tool could facilitate automatic stylistic transformation of a text toward an obscure style, or “away” from the stylistic distinctiveness of the author. It’s possible that the same processes could be reversed to reconstitute the original style of a text. Hence a style encryption/decryption system could be created.

1.2.3 Style based search

Style based search proposes a new kind of mass-document search (such as Internet searching) where instead of supplying words and phrases as search terms, one uploads an entire document and the engine returns other documents of similar style within certain tolerance levels.

1.2.4 Educational and productivity tools

Productivity software titles like Microsoft Word increasingly integrate grammar and style tools to aid in writing and suggest paraphrases to the user. With a style transformation tool such as the one we are proposing, the user could be shown more options and flexibility in re-writing suggestions. Instead of evaluating along just one axis (the generic “good writing style” that MS Word tools are coded for), the user could experiment with multiple known styles, or derive a new style from a corpus examined by the tool.
1.2.5 Authoring tools and digital entertainment applications

In addition to the productivity software functionality, there could also be more sophisticated author-centric tools, taking advantage of stylistic transformations that could aid in development of characters, narration and expository passages. Authors could get ideas by automatically paraphrasing their writing along stylistic dimensions. Characters in a dialogue could acquire different personalities detectable by their speaking styles. We see application to digital entertainment whereby written passages or character utterances could be dynamically transformed to some desired target style using some of our methods. This application could possibly free up writers and designers from having to rewrite multiple versions of the same text in different styles.

1.2.6 Machine Translation Enhancement

One of the problems inherent in Machine Translation (MT), and Statistical Machine Translation (SMT) is the stylistic differences present in various languages. After (an often literal) translation, the resulting text may be intelligible but still not in the form that the speakers of the target language are familiar with. A robust transformation process, for example, between “post-MT style” and “native style,” could solve this problem.

1.2.7 Interactive agents

Robots and digital agents that interact with human users through text (or speech directly synthesized from text) have often hard-coded communication styles. To take a simple example, Microsoft Office’s “clippy” was an interactive help agent that appeared in strategic moments and attempted to solve a perceived problem through a kind of text-based dialogue with the user. We could imagine more sophisticated agents with unique personalities each communicating in different styles. Ideally, such agents could be trained to morph their communication styles to one best suited for the user. The training could come from analyzing user’s existing writing pieces or emails. A stylistic profile of the user could be derived that could form the basis for the agent’s personality, or at the very least automatically select the best choice among multiple static personalities supplied.
1.3 Related work by others

Abstract

Text style is difficult to define and indeed has no precise enough definition for computational purposes. In this section, we begin by familiarizing the reader with types of text classifications available. Next we discuss notions of style as defined throughout the literature space in multiple disciplines. We further discuss style markers used in mainly classification or text analysis research in order to provide a base for using them in our system. This section can be viewed as a survey of the research literature related to text styles and text classification work.

1.3.1 Text classifications: genre, text type and register

In [Moe01] a good background is given on how text type, genre and style relate to each other in linguistics. The author begins by describing genre as non-linguistic classification of written work in the tradition of Aristotle. When the designers of the Helsinki Corpus [Ris94] wanted a definitive classification of all English text, they found traditional genre, as used in the field of literary studies, inadequate. They felt they had to add classes of non-literary texts such as “letter, proceeding, trial” [Moe01] in addition to a catch-all “none of the above” category. Thus [Ris94] called their classification variable “text type” rather than genre.

Both text type and genre are non-linguistic classifications, but each of their classes exhibit high correlation to one or more linguistic features. As [Moe01] describes:

Examples of such correlations are that narrative text types contain past tense verb forms; biographies are written in the third person singular, etc. The set of linguistic features which characterizes a particular text type or genre is referred to as the text type style or genre style. Some linguists also use the expression ‘register’ with this meaning.

Biber was able to derive purely linguistic text types by aggregating and clustering linguistic features. In [Bib89], Biber identified 67 features, based on previous studies that had associated them with one or more genres. Through calculating various combinations of co-occurrence frequencies of these features, Biber produced 8 “text type” categories derived using automatic clustering algorithms.
1.3.2 Style and Stylistics

Andreas Jucker [Juc92] likened “style” to what de Sausser calls “parole” and what Chomsky calls “performance.” Jucker calls style a “comparative concept” describing relative differences between “texts or discourses” and in some cases between text/discourse and “some kind of explicit or implicit norm” [Juc92].

He also contrasts the popular view of the term “style” with the notion in traditional stylistics:

*There is a lay notion of the concept of style, which equates style with the elevated and aesthetically pleasing forms that are used, for instance, by celebrated authors in their writings. Some newspapers, accordingly, are claimed to “lack style” altogether.*

*This is of course not what traditional stylistics takes style to be. Every single text has got a style as far as it has formal properties that can be compared with those from other texts. A stylistic analysis will try to single out those features that help to distinguish the texts under comparison. One particular feature may occur in only one text and not the other, or it may appear with a frequency that is appreciably different from one text to the other.* [Juc92, page 12]

Stylistics has been called a conceptual successor to the ancient Greek concept of rhetoric [Bib89].

Walpole calls stylistics an “offshoot of linguistics” which “takes the position” that “style is a deviation from normal language usage” [Wal80].

[Kar04] contextualizes style with this definition:

*A style is a consistent and distinguishable tendency to make some [of these] linguistic choices. Style is on the surface level, very obviously detectable as the choices between items in a vocabulary, between types of syntactical constructions, between the various ways a text can be woven from the material it is made of.* [Kar04]

Fish in his essay “What is stylistics and why are they saying such terrible things about it,” called stylistics simply the “application of linguistics to the study of literature” [Fish81]. Simpson writes in *Stylistics: A resource Book for Students* that “Stylistics is a method of textual interpretation in which primacy of place is assigned to language” [Sim04]. George Heidorn of Microsoft is probably closer to the lay definition of “style”, when he defines “style checking” as follows.
Style checking refers to checking text for errors that a book on good writing style, would discuss, such as overuse of the passive voice. [Hei00]

Whitelaw and Argemon in [Whi04] propose a definition for “stylistic meaning:”

_We provisionally define stylistic meaning of a text to be those aspects of its meaning that are non-denotational, i.e., independent of the objects and events to which the text refers [Whi04]._

This definition raises one of the central questions concerning style: style’s relationship with other concepts such as meaning or personality. While it may be more functional to think of style as inherently separate from content, personality and meaning, or only affecting the manner of delivery rather than the message itself, most experts agree that style is not disjoint from meaning entirely.

As far back as 1753, Geroge-Louis Leclerc de Buffon declared that “Style is the man himself,” [Buf21] implying an inseparability between style and personality. While this contention may form a solid theoretical basis for authorship attribution that we examine below, it does suggest mechanical style transformation may not be possible. Similarly, John Middleton Murry wrote that “style is not an isolable quality of writing; it is writing itself” [Mur22]. Simpson admits that style affects meaning, but it is not meaning in its entirety:

_While linguistic features do not of themselves constitute a text’s ‘meaning’, an account of linguistic features nonetheless serve to ground a stylistic interpretation and to help explain why, for the analyst, certain types of meaning are possible. These ‘extra-linguistic’ parameters are inextricably tied up with the way a text ‘means’. The more complete and context-sensitive the description of language, then the fuller the stylistic analysis that accrues. [Sim04, page 2]_

Walpole suggests that style and discourse may not be so easily separable when she writes: “We no longer feel comfortable viewing style as the dress of discourse, the external and changeable garb for ideas…” [Wal80]. However, as discovery write below, Walpole’s model of “style as option” does allow for extra-linguistic analysis and even the possibility of style transformation.

### 1.3.3 Stylistics in humanities

Richard Bradford in [Bra97] describes stylistics as a conceptual descendent of the ancient Greek “rhetoric.” During the early part of the 20th century, two different and disjointed groups were carrying on literary criticism focusing on what can be called “stylistics.” The first was the Russian/Central European formalists. The other group was the British and American teachers who practiced what later became known as “New Criticism.” The two groups began cooperating during the 1960’s and developing overlapping goals and methods. After the 60’s, however, both groups had their academic predominance “unsettled” by a wave of interdisciplinary practices. Structuralism, post-structuralism,
feminism and new historicism, all became significant elements of contemporary literary studies.

Bradford describes two new groups emerging in the last years of the 1970’s called “textualists” and “contextualists.” The textualist group is comprised of New Critics and the European Formalists because “they regard the stylistics features of a particular literary text as productive of an empirical unity and completeness” [Bra97]. The contextualist group is comprised of the elements involved in the 60’s movements such as post-structuralism, feminism and Marxist analysis. This type of stylistics involves “a far more loose and disparate collection of methods” [Bra97] but its members are unified in their concentration on the relation between text and context.

1.3.4 Reactions to stylistic oriented methods

Some commentators in a variety of disciplines do not consider stylistics as a legitimate or fruitful method of analysis. The main criticisms seem to be about the perceived arbitrariness of the meaning-assignment to stylistic methods and interpretation of algorithmically generated stylistic data. A certain stereotype of who stylisticians are and how they operate is often embedded in the criticism.

Simpson summarizes the attitudes in this 2004 book:

There appears to be a belief in many literary critical circles that a stylistician is simply a dull old grammarian who spends rather too much time on such trivial pursuits as counting the nouns and verbs in literary texts. Once counted those nouns and verbs for the basis of the stylistician’s ‘insight’, although this stylistic insight ultimately proves no more far-reaching than an insight reached by simply intuiting from the text.” [Sim04, page 2]

Fish situates stylistics as “a reaction to the subjectivity and imprecision of literary studies” [Fis81]. He does not credit this reaction with adding any additional understanding, however:

The machinery of categorization and classification merely provides momentary pigeonholes for the constituents of a text, constituents which are then retrieved and reassembled into exactly the form they previously had. There is in short no gain in understanding: the procedure has been executed, but it hasn’t gotten you anywhere. Stylisticians, however, are determined to get somewhere... [Fis81, page 55]

In addition, the profiling function, or emergence of stylistic categories which is so central to stylistics is also attacked by Fish:

The establishment of a syntax-personality or of any other paradigm is an impossible goal, which, because it is also an assumption, invalidates the procedures of the stylisticians before they begin, dooming them to successes that are meaningless because they are so easy.” [Fis81, page 56]
Freeman reserves some strong criticism, what he calls the “heart of my quarrel with stylisticians,” to the arbitrariness of what text features are or should be considered “significant.” He says of the stylisticians:

\[...
\]

For Fish stylisticians also commit the de-humanizing sin of creating meaning where none exists, at least not according to any existing literary theories. The new meaning assumptions become a circular justification for the utilization of stylistic methods:

\[...
\]

Fish also offers a psychological explanation for the stylisticians’ behavior.

\[...
\]

While not getting quite as personal as Freeman, Carter and Simpson in [Car88] warn that stylistic analysis is too limiting since it has no access to the “extra-textual world of social, political, psychological, or historical forces” [Car88, page 7]. Carter and Simpson acknowledge stylistic trends to take such forces into account, but:

\[...
\]
Another important attack came from linguist Jean-Jacques Lecercle in 1993 and was recounted in the introduction of the 2004 book *Stylistics: A Resource Book for Students*:

> Some years ago, the well-known linguist Jean-Jacques Lercercle published a short but damning critique of the aims, methods and rationale of contemporary stylistics. His attack on the discipline, and by implication the entire endeavor of the present book, was uncompromising. According to Lecercle, nobody has ever really known what the term “stylistics” means, and in any case, hardly anyone seems to care. Stylistics is “ailing”; it is “on the wane”; and its heyday, alongside that of structuralism, has faded to but a distant memory. More alarming again, few university students are “eager to declare an intention to do research in stylistics” [Sim04, page 2].

The criticisms leveled at stylistics seem to concentrate on the subjective aspect of “interpretation” based on statistical data. One may be tempted to think that one could escape this criticism to a great extent in certain areas like computational linguistics, by simply not engaging in any interpretation of the results. However, [Car88] suggests this is not so easily achieved, if even possible:

> The assignment of meaning or stylistic function to a formal category in the language remains an interpretive act and thus cannot transcend the individual human subject who originates the interpretation. Thus, while the recognition of specific formal features can in most cases by attested within the terms of the system, the analyst has to be taken on trust in his or her interpretive assignment. It is a perennial problem, or even dilemma, in stylistics that no reliable criteria can be generated whereby specific functions or effects can be unambiguously attributed to specific formal features of the language system. [Car88, page 6]

This seems to suggest that even selection of language features to study and assigning of literal or figurative value to them could be considered problematic. But what if one was to simply describe the findings without making any assignment to “specific formal features”? Even there, [Car88] says that one cannot escape “interpretation:”

> Any resolution to describe the data rather than interpret it constitutes an interpretation. It is an interpretation of the way literary study can or should be approached and analysis of it conducted. [Car88, page 6-7]

### 1.3.5 Defense of stylistics

While having many skeptics in linguistics and literary criticism, there are those who have embraced and continue to advance the discipline of stylistics.

Simpson in [Sim04] answers Lecercle’s attacks on stylistics:

> Modern stylistics is positively flourishing witnessed in a proliferation of sub-discipliens where stylistic methods are enriched and enabled by theories of
discourse, culture and society: feminist stylistics, cognitive stylistics and discourse stylistics are established branches of stylistics which have been sustained by insights from, respectively, feminist theory, cognitive psychology and discourse analysis. [Sim04]

Simpson suggests that stylistic methods can best be defended when the “three R’s” are observed: “Stylistic analysis should be rigorous, retrievable, replicable” [Sim04].

Walpole also offers redeeming values of stylistic methods and at the same time demystifies the statistical aspects for the liberal arts researchers. She quotes Corbett:

> Some instructors try to heighten an awareness of style by a detailed scrutiny of published prose. As Edward Corbett suggests, "Any stylistic analysis must start out with some close observation of what actually appears on the printed page." This observation includes counting sentence lengths, types and numbers of syntactic structures, and classes and varieties of words—what De Quincey called the "mechanology of style." Classroom applications of these analytic approaches can be found in professional journals. "Such a procedure would make counters and measurers of us all," says Corbett, admitting that this may seem repellent to humanistically-trained teachers; "but this is a necessary step if we are to learn something about style in general and style in particular." For it gives us the requisite things to point to as we test the effect of authorial decisions.

Giacomo Ferrari in [Fer03] justifies the move toward empirical and statistical methods because of the inadequacy of relevant linguistic theory. Ferrari offers the move toward computational linguistics as a natural result of linguistics’ failures to offer better and more comprehensive models both in syntax and semantics. Regarding syntax, he writes:

> Talking of syntactic approaches, we observe that in many cases the move toward empirical studies is uncontrolled, because linguistics has no interpretation mechanisms. Computational Linguistics produced important results in syntactic and morphological analysis because it could rely on robust theoretical linguistic models like Chomsky’s grammar (s) and the theory of automata. This is not true in other fields like, for example, discourse and dialogue modeling. [Fer03, page 163]

And here, he makes a similar criticism in semantics and suggests that linguistics theory should more fully embrace computational linguistics.

> In the field of semantics, Computational Linguistics turned to logic because linguistics offered absolutely nothing for what concerns the meaning of sentences. Thus the conclusion is that Linguistics is in debt here, and should take advantage of stimuli coming from Computational Linguistics to build new theories which take into account computational modeling. [Fer03, page 163]

What the above researchers tell is that not only is stylistics a viable and growing discipline, but its models do not necessarily produce meaning or make claims beyond their own
symbolic system. Stylistics thus need not be called a meaning-producing enterprise, but perhaps a hypothesis-checking enterprise validating or invalidating human theories of language.

Still a notion of style as a distinguishable “thing” is necessary in order to work with it, and the problems with overlapping domains of style and meaning, or style and personality remain.

1.3.6 Style as “option”

In “Style as Option,” [Wal80] Walpole tackles the aforementioned problem by stressing a conception of style as “choices among alternatives.” She divides these choices into two groups: linguistic and extra-linguistic.

The framework represents perhaps the best operational definition for style that would allow robust models to achieve solutions to narrow problems involving style.

Walpole explains “What remains of prose after we have set aside the detachable ideas and the immutable requirements is Style, the vast area of writer's choice.” She elaborates on the omnipresence of options in writing:

*Options exist in small matters: the individual words, the optional comma that influences emphasis, the placement of a movable adverb, the different rhythmical effects of under and beneath. Options exist in sentence variations: coordination vs. subordination, clausal vs. phrasal constructions, polysyndeton vs. asyndeton. Options exist in larger decisions: whether to use parallel sentences, whether to explain a point in depth or superficially, whether to illustrate a concept with several short examples or one long analogy. And options exist in the total work: the level of diction, the attitude toward the audience, the weight given to flourishes or simplicity. Though writers may be constrained by imposed limitations, they have in each of these areas wide freedom of selection. And in each of these areas they produce, in Ross Winterowd's phrase, “features that we can 'point to.'” [Wal80]*

These features we “can point to” are perfect candidates for “style markers” that we can use to assess the presence of that style.

Walpole also explains that style as “option” necessitates two assumptions. First, that the style is actually a conscious choice by the author and second, that it is a variable quality not necessarily always present with the same intensity and “not a unique and inseparable reflection of the author's (or student's) personality” [Wal80].

This view of style accommodates the every day observation that writers can and often do deliberately change their own style. A view of style “inseparable” from writing would not accommodate this. At the same time, a component of these choices may be unconscious. This, we think, Walpole would call an extension of personality rather than style.
Walpole’s own background in pedagogy allows for more unique observations allowing us to entertain the possibility of altering and transforming written style.

Literature students are already doing a form of stylistic transformation exercise, as Walpole observes:

\[\text{A second approach sometimes followed involves exercises of deliberate imitation.\ Corbett again provides guidance in this traditional technique. Students learn how to choose words and structures that echo the choices of an acknowledged style master. As they clothe their own matter in a borrowed manner, they are sensitized to the range and impact of writing options [Wal80].}\]

She describes another exercise where students are encouraged to highlight and later subdue “obvious style” [Wal80].

1.3.7 Style markers and computational stylistics

Style markers, also referred to as “features,” at their most basic, are indications of choices that were made in writing in a recognizable way. A uniformed and extensible set of markers, with well defined scopes are necessary for an automatic style analyzer. One objective of our project is to publish a comprehensive list of such markers. We surveyed the literature in order to compile a list of existing markers used for various related purposes in stylistics.

Various works have offered examples of, as well as process suggestions for, working with style markers. DeQuincey’s notion of “mechanology of style,” includes “counting sentence lengths, types and numbers of syntactic structures, and classes and varieties of words” [Wal80].

Heidorn in [Hei00] briefly discusses the style-checker component of Microsoft Word 97. That commercial product uses a set of 21 or so “grammar and style options,” each of which is a rule that checks for a specific stylistic phenomenon, for example “correct capitalization,” and “misused words.” Microsoft has coded these rules developed by their team of linguists.

Almost all the options operate at sentence level. The Microsoft notion of “style” here is a single dimensional discrete variable. At one end, “formal”, all 21 options are checked for. At the other end, “casual,” none is checked for. The MS-Word 2002 product, a successor to the one discussed in [Hei00] uses 22 distinct options for style. Many like clichés, contractions, gender-specific words, use of first person, numbers, sentences beginning with “And,” “But,” and “Hopefully” use pre defined assets that they check for. Others such as “wordiness”, “sentence length”, “hyphenated and compound words” and “successive nouns/propositional phrases” do procedural checking using NLP tools.

Lyuckx and Daelemans identified four types of features in [Luy04]:

---

18
1. **Token-level** such as word lengths, syllables and n-grams
2. **Syntax based** like part-of-speech and rewrite rules
3. **Vocabulary richness**: type-token ratio, hapax legomena
4. **Common word frequencies**

They also made the additional observation that “most studies are based on word forms and frequencies in occurrence” [Luy04].

An interesting example of (2) from above may be the Brill Tagger [Bri04]. Eric Brill and his team at Microsoft developed a two tier system whereby a relatively dumb part-of-speech (POS) tagger initially tags all words based on their statistically known most likely POS. Subsequent routines check for specific patterns and alter the tags if they find them. The routine uses variables that stand in for words three lexemes ahead and three behind the current word. Rules are specified in terms of these variables and the current word-tag tuple being considered for alteration [Bri04].

According to [Arg03], “stylometric models have been based on hand-selected sets of content-independent, lexical, syntactic or complexity-based features.” In [Juc02] the authors discuss how to proceed after a given super-set of style markers is provided. Three specific points are especially of note. A “stylistic analysis” phase should concentrate on the best features among all that is available based on which ones exaggerate the difference between the two texts being compared. Second, any frequencies have to be seen in relation to the length of the text so that “we should talk in terms of density, that is to say the frequencies of a feature within a well-defined stretch of text” [Juc02]. Lastly, the authors repeat a point from Enkvist in 1980 suggesting that the “guiding principle” in comparison of feature sets should be to keep as many non-linguistic features as possible constant over all the texts to be compared in order to be able to assign the linguistic difference with more confidence to those few features that do vary. [Juc02]

Latent Semantic Analysis [Lan98] and Probabilistic Latent Semantic Analysis [Lat08] are techniques used to determine word relationships. In these techniques a larger set of initial distinct tokens may be reduced as more words are found to be equivalent on semantic grounds.

Currently many writing analysis techniques are used in natural language processing, computational linguistics and readability assessment. WordNet, an online word ontology tool, can be used to calculate statistics of different word groupings. The Gunning fog index [Haa07], and the Flesch-Kincaid readability test [Haa07] are two popular means of assigning reading level by analyzing vocabulary in writing.

Authorship attribution works from Khosmood and Levinson [Kho05][Kho06] contain an enumerated set of criteria used as features. Their markers include vocabulary measurements [Fak01], lexeme statistics [Kes03], as well as statistics based on grammatical constructs.
1.4 Related work by us

Abstract
In this section we briefly describe our research related to this project in more depth, beginning with work in source attribution in 2005 and ending with recent publications on a style processing system in 2008.

1.4.1 Research in source attribution (classification)

In “Automatic Source Attribution of Text: A Neural Networks Approach” [Kho05], we advanced some arguments in favor of the term “source attribution” instead of “authorship attribution.” This was mainly due to the fact that the term “authorship” was limiting in describing the potential of the technologies, as well as the fact that none of the features extracted from texts could be definitively called an indication of “authorship.” The term “source,” we felt was a better designation capturing situations between different authorial styles as well as institutional sources such as movie scripts, songs, White House announcements where multiple authors had authorial input. We demonstrated this by doing successful neural network based text classification by using the same process first classifying different corpora based on author and then based on genre within the work of a single author (Shakespeare). We were able to achieve a high level of accuracy (70%) distinguishing Shakespearian comedies from tragedies using non-trivial, in-common word frequencies.

In “Toward Unification of Source Attribution Processes and Techniques” [Kho06], we explored the idea of diverse classification methods all working toward the same goal with unified feature selection and common abstraction layer for classification methods. We performed source attribution with a standard corpus using two different methods (Naive Bayes and Nearest Neighbor) with two different feature sets (phrases, bi-grams).

1.4.2 Research in style processing (classification and transformation)

Beginning with an NSF proposal submitted in November of 2006, we began considering the idea of style-based text transformation, some independent study research into this and related areas followed over the next few quarters. A literature review (some of which appears in this document) and other reports were produced.

In 2008, we presented a paper [Kho08-2] at the BCS Corpus Profiling Workshop on automatic style classification and transformation largely outlining the system we discussed here. We show a transformation problem with a simplified system of 10 markers and 3 transforms. The markers are listed in figure 1. The three transforms are:
- T1: Removes any mid-word hyphens and creates two words where there was one hyphenated word before, for example: {“single-minded” → “single minded”}
- T2: Substitutes word replacement suggestions given in the database in GNU dictio.
- T3: Expands acronyms related to US Government.

The source text is a legal declaration from US Department of Justice and the target style profile was that of George Orwell’s Animal Farm. After applying the three transforms, we show that the distance metric between the source and target texts reduces. Visually, however, there is not much at all that a human expert could identify as “Orwell” or “Animal Farm” in the transformed text.

<table>
<thead>
<tr>
<th>Marker</th>
<th>Target</th>
<th>Source</th>
<th>S.T1</th>
<th>S.T2</th>
<th>S.T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg S per P</td>
<td>5.85700</td>
<td>2.66667</td>
<td>2.66667</td>
<td>2.66667</td>
<td>2.66667</td>
</tr>
<tr>
<td>avg W per P</td>
<td>103.43700</td>
<td>77.00000</td>
<td>77.33330</td>
<td>77.33330</td>
<td>81.33330</td>
</tr>
<tr>
<td>avg C per P</td>
<td>455.30300</td>
<td>428.33300</td>
<td>428.00000</td>
<td>427.00000</td>
<td>460.33330</td>
</tr>
<tr>
<td>avg W per S</td>
<td>17.66000</td>
<td>28.37500</td>
<td>29.00000</td>
<td>29.00000</td>
<td>30.50000</td>
</tr>
<tr>
<td>avg C per W</td>
<td>4.40200</td>
<td>5.52777</td>
<td>5.55345</td>
<td>5.52155</td>
<td>5.65980</td>
</tr>
<tr>
<td>avg Sy per W</td>
<td>1.33000</td>
<td>1.77000</td>
<td>1.77000</td>
<td>1.75000</td>
<td>1.77000</td>
</tr>
<tr>
<td>avg frequency of words</td>
<td>0.00042</td>
<td>0.00794</td>
<td>0.00787</td>
<td>0.00787</td>
<td>0.00773</td>
</tr>
<tr>
<td>ratio of W &gt; 6 /W</td>
<td>0.02144</td>
<td>0.02165</td>
<td>0.02155</td>
<td>0.02596</td>
<td>0.02459</td>
</tr>
<tr>
<td>ratio of W &gt; 2 Syl</td>
<td>0.09530</td>
<td>0.10303</td>
<td>0.29741</td>
<td>0.29319</td>
<td>0.31147</td>
</tr>
<tr>
<td>Linus dictionary hits</td>
<td>0.97963</td>
<td>0.99206</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
</tr>
<tr>
<td>RMS Error versus Target</td>
<td>n/a</td>
<td>5.76930</td>
<td>5.72002</td>
<td>5.71864</td>
<td>5.63488</td>
</tr>
</tbody>
</table>

Table 1, 10 style markers [Kho08-2] and their measured presence after application of each transform (T1, T2, T3) to the source (S) serially.

Figure 1, RMSE based distance in [Kho08-2] after serial application of transforms T1, T2 and T3 to S.
This result is probably because there are only 3 very simple transforms. We hypothesize much more powerful transforms and many more markers are needed to achieve human recognition of this stylistic shift.

A related paper [Kho08-1] has been accepted at WCECS-08 and another abstract has been accepted for the DGfS 2009 “Workshop on Corpus, Colligation, Register Variation” at Universität Osnabrück, Germany.
2 System, tools and examples

2.1 Stylistic classification-transformation system

Abstract
In this section, we lay out the designs for an automatic stylistic classification and transformation system. We begin by repeating the goals and enumerating the requirements for such a system. Next we delve into the system design itself and explain each subcomponent in terms of communications interfaces and function. This is followed by a discussion on implementation technologies and tools. Finally a few theoretical examples are provided to demonstrate a fairly advanced goal state for the system performance. These examples should help crystallize the transform functionality of the system.

2.1.1 Goals

The major goal of building this system is to demonstrate the ideas about style specification, stylistic classification, stylistic transformation and to advance the research in computational stylistics. There are really two large phases to this development. The first is the “engine” or the core part of the program which is able to classify and apply transforms given a superset to style markers and transformation operators. In addition, the system allows for the use of multiple and flexible evaluation algorithms, such that researchers could experiment with different algorithms and parameters.

The second part of the project is an open ended process of specifying style markers, style transformation operators and evaluation algorithms. We think that diverse techniques and intended subjects and general thinking behind each of these components can contribute to creative and unexpected solutions. It is our intention that the system store and utilize a growing set of these methods and specifications to use with future problems.

We intend to create clean, logical and web-driven interfaces, to make it easy for researchers to run experiments and contribute to the collections.
2.1.2 Requirements

The functional requirements of the system can be stated more concretely as follows:

1. System is able to accept, test and store style markers from users
   a. The presence of each marker is calculated using a marker-detector function that accepts either a sentence or a paragraph or an entire document as a parameter.
   b. The marker-detector function returns a single floating point value between 0 and 1, denoting the level of presence of a marker in the text.
   c. Each function can analyze the input text for patterns of tokens, lexemes, and grammatical constructions, presence of a select set of vocabulary or other criteria.

2. System is able to accept, test and store style transformation operators
   a. For our purposes, a transformation operator is a function that converts one or more sentences to other sentences, trying as much as possible, to preserve the a priori meaning of the text.
   b. A transform operator accepts either a sentence, or a paragraph or the entire document as input and returns the same selection as output.
   c. Each transformation operator is written for a different and specific purpose and may be associated with one or more target corpora, and hence excluded for transformations not involving those corpora.

3. System is able to accept and use a user supplied evaluation function
   a. An Evaluation function accepts two arrays of marker values one representing the derived analysis of the target corpus and the other representing the analysis from the source text (text being transformed).
   b. The function itself is upload-able for work within the system.

4. System is able to perform classification of text
   a. Classification is done based on the evaluation function and determines a closest match with respect to previously classified corpora within tolerance levels, or “no match.”
   b. A new corpus can be classified by the system, but if it falls within the tolerance levels of another previously classified corpus, the system should signal the operator that there is possible under-specification of styles.

5. System is able to perform transformation of text
   a. Transformation function is a planner that applies one or more operators to a source text in order to minimize the metric distance between source and target texts.
   b. Transformation function is optionally capable of minimizing reuse of the same operators, and minimizing use of operators on the same sentence.
In addition, given the open source nature of the system and its intended community based use, we add the following non-functional requirements:

1. All functions of the system are accessible via the World Wide Web, including uploading of corpora and source texts for analysis.
2. The system uses a database to store style markers, transform operators, function and previous corpus analysis for future use and analysis. Each submission will be associated with metadata denoting credit information and intended use of the uploaded module.
3. The system provides meaningful output for classification and transformation functions.

2.1.3 Design

Given the constraints from the above requirements, we present the following design for the function part of the system. Each section will be listed in alphabetical order and described in detail below in figure 2.

- **Analyze source**: This module executes all the relevant style marker-detector functions and stores the resulting returns in the source style array or $S_0$ vector. Each subsequent modification to the source style will be stored in a separate array such as $S_1$, $S_2$, $S_3$, etc.
• **Analyze target**: This piece executes all the relevant style marker-detector functions and stores the resulting returns in the target style array.

• **Apply transforms**: This piece draws one operator or a group of operators from the operator library and applies them to the current S text. It also has the capability to “backtrack” by applying the next transform not to the current S text, but to any version in the past as well.

• **Classify**: In classification mode, the system associates the source text with an existing corpus along stylistic dimensions.

• **Compare**: This is a simple step where the source and target analysis arrays are prepared for evaluation.

• **Evaluator**: The user-defined algorithm that provides a distance between source and target and also decides if that distance constitutes “matching”.

• **Operator library**: A collection of operators in a database, some with restrictions as to the corpora they can be associated with. The system draws an appropriate operator or group of operators in the “Apply Transforms” stage.

• **Target corpus**: A corpus is made up one or more documents which together constitute as a text exhibiting a unique style. All plays of Shakespeare, for example would be the “Shakespeare corpus” being made up of tens of documents.

• **Transformed text**: The output of the transformation function, this text is generated either when the metric indicates that it is a stylistic match with the target (“matched”), or when the operator library is exhausted and it is the latest transformed S version (“best effort”).

### 2.1.4 Transformation operators and transforms

Transformation operators are atomic actions that alter text in some way. We call a collection of operators with a certain common characteristics, a transform.

The transforms discussed in the above Shakespeare example were not fully defined. The ones it he 2-marker example were too simplistic. In order to provide a better appreciation of the depth and breadth of transformation operators and the infrastructure necessary for them to work, we will discuss these in more detail here.

#### 2.1.4.1 String replacement operators

These are operators that replace one string with another from a dictionary of equivalences. For example, the pronoun form replacement when modernizing Shakespeare {“thou” → “you”} would be a replacement operator. Another example would be acronym expansion like {“DMV” → “Department of Motor Vehicles”}. A third example would be vocabulary replacement such as {“utilize” → “use”}. Building on string replacement operators, lexeme replacement transforms are possible.
Relying on string replacement operators means we will quickly begin to encounter problems of ambiguity. There are for example multiple classes of acronyms for different subjects, or multiple dictionaries for different purposes. Thus, we will have to facilitate operators with additional constraints as pre-conditions and post-conditions. Pre-conditions are generally the high probability of presence of marker or markers as measured by our marker-detector algorithms for the classification function. Post conditions are variables indicating execution parameters of the transform. These post conditions can be used by other operators as pre-conditions to satisfy their constraints or global constrains such as avoiding excessive operations on the same section of text.

Surface replacement operators generally work on situations of little word-sense ambiguity. However, word-sense requirements can be easily adapted as additional constraints in operator pre-conditions. Senses can be determined either by linguistic parsing or by context analysis. Operator constraints based on a measure of confidence in correct sense resolution are possible.

2.1.4.2 Semantic operators

This class of operators can be said to be “semantically aware” of the text to be operated on. In general full or partial parsing of sentences is necessary in order to achieve the desired transformation. Since parsing of free form English itself is far from a solved problem, a context sensitive threshold of “parse-ability” can serve as a useful constraint in applying the operator. Link Grammar Parser provides such a feedback mechanism.

As an example of a semantic operator, we consider the following line from Winston Churchill:

*Let us therefore brace ourselves to our duties, and so bear ourselves that if the British Empire and its Commonwealth last for a thousand years, men will still say, this was their finest hour.*

We then use a known algorithm [Yod08] to convert it to “Yodaspeak” (i.e. as the character “Yoda” from Star Wars would say it):

*Therefore brace ourselves to our duties, let us, and for a thousand years so bear ourselves that if the British Empire and its commonwealth last, still say, men will, their finest hour, this was.*
2.2 Sample walkthroughs

Abstract
In this section we discuss in depth a number of examples of various complexities denoting the kind of work a fully functional system could perform. It is important to note that all scenarios in this section are hypothetical as a fully functioning system does not yet exist to perform them. However, given some assumptions, they should be practical to support for the final project.

2.2.1 Modernizing Shakespeare

In this example, we demonstrate how a stylistic processing system could work to convert a few lines of Shakespeare to a more modern style.

Source S is in the style of Shakespearian Early Modern English, as profiled by Act I of Shakespearian play Hamlet. Style T is Modern American English as profiled by processing corpora of literary works in the 20th century written in modern American English. Several full length novels by Tim O’Brien, Don DeLillo and Toni Morrison should be enough to profile the style.

Marker-detector M1 measures the density of modern English pronouns in a piece of text. Style T, consisting of modern works, has more occurrences of these pronouns. Thus the relation M1(T) > M1(S) holds at the outset.

Operator P1 modernizes archaic pronoun forms such as “yea”, “thou” or “thine”, through strict lexeme substitution.

Sentence A, a part of S, is this line from the Shakespearian play Hamlet:

\[
\begin{align*}
This & above all: to thine own self be true, \\
And it must follow, as the night the day, \\
Thou canst not then be false to any man.
\end{align*}
\]
Operator P1 is applied to sentence A. Under specific implementation choices, P1(A) results in this sentence:

This above all: to your own self be true,
And it must follow, as the night the day,
You canst not then be false to any man.

B is assigned to the outcome, i.e. B=P1(A).

This operation yields the following: M1(B) > M1(A) which means the text is now closer to target style T along the dimension measured by M1 marker. A whole family of such language-specific operators can be employed in the same fashion, maximizing other metrics measuring modernity of language use.

After successive applications of many transformer functions, the result may be like this for B:

This is the most important of all: be honest with yourself,
and then it follows, just like the night follows the day,
that you cannot be dishonest to anyone.

In addition we wish to maximize a new metric, M2, which measures the average number of words per sentence in a piece of text. It holds that M2(s) > M2(t). Operator P2 is designed to split large sentences into two, whenever possible. Applying P2(B) would yield the following:

This is the most important of all. Be honest with yourself. And then it follows, just like the night follows the day, that you cannot be dishonest to anyone.

Operator P2 split the sentence into 3 sentences by simply converting colons and conjunctions to sentence boundaries. The result is that M2(B) > M2(P2(B)), in other words, the transformed source has moved closer to the target style after applying two transformations.
2.2.2 A trivial 2-marker scenario in depth

In this scenario, we manually follow the actual steps in the system for a trivial transformation. We are given a corpus, \( T \), which can be described as modern, formal, factual writing with short sentences. We only have two markers in this scenario, specified with two corresponding marker-detector functions, and two transform functions. The marker-detector functions are as follows:

1. \( M1() \) returns inverse of the average words per sentence (i.e. sentence/word).
2. \( M2() \) returns a ratio of overall instances of single digit numbers written using Arabic numeral symbols (i.e. “1”, “2”, etc.), to the total number of single digit numbers written in the text.

We are also given 2 transform functions:

1. \( X1() \) uses some symbolic linguistic techniques to split large sentences into two or more smaller ones.
2. \( X2() \) replaces instances of a number written in English to its equivalent in Arabic numeral symbols.

Finally, we are given the following text from a recent new story as source text (S):

> Canada's opposition parties have reached a tentative deal to form a coalition that would replace Prime Minister Stephen Harper's minority Conservative government less than two months after its reelection, a senior negotiator said on Monday.

This text contains 35 words and one of them is a number (“two”).

A Simplified transformation run through the system is the following:

1. Analyze the target corpus, \( T \) (for this example we specify target corpus characteristics)
   a. \( M1(T) = 0.2 \)
   b. \( M2(T) = 0.8 \)
2. Analyze the source text, \( S \)
   a. \( M1(S) = (1/35) = 0.0286 \)
   b. \( M2(S) = (0/1) = 0.0 \)
3. For comparison and evaluation, a simple root mean squared error (RMSE) is calculated between the \( S \) and \( T \) vectors.
   a. \( \text{RMSE}(S,T) = \sqrt{(0.2 - 0.0286)^2 + (0.8 - 0)^2} = 0.8182 \)
   b. The system determines the RMSE distance is too high to constitute a match
4. Operator \( X1 \) is applied to source text, \( S \)
   a. Operator \( X1(S) \) converts the text to version \( S_2 \) (inserted words and punctuation are underlined)

> Canada's opposition parties have reached a tentative deal. The deal is to form a coalition. The coalition would replace Prime Minister Stephen Harper's minority Conservative government less than two months after its reelection, a senior negotiator said on Monday.
government. This is less than two months after its reelection. A senior negotiator said it on Monday.

5. S₂ is analyzed
   a. \( M1(S_2) = ((1/8)+(1/7)+(1/11)+(1/9)+(1/7))/5 = 0.1225 \)
   b. \( M2(S_2) = (0/1) = 0.0 \)

6. New RMSE is calculated
   a. \( \text{RMSE}(S_2,T) = \sqrt{(0.2 - 0.1225)^2 + (0.8 - 0)^2} = 0.8037 \)
   b. System determines RMSE is still too high so we continue.

7. Operator X2 is applies to source text, S
   a. Operator X2(S₂) converts the single instance of “two” to “2” and produces S₃.

   Canada's opposition parties have reached a tentative deal. The deal is to form a coalition. The coalition would replace Prime Minister Stephen Harper's minority Conservative government. This is less than 2 months after its reelection. A senior negotiator said it on Monday.

8. S₃ is analyzed
   a. \( M1(S_3) = ((1/8)+(1/7)+(1/11)+(1/9)+(1/7))/5 = 0.1225 \)
   b. \( M2(S_3) = (1/1) = 1.0 \)

9. New RMSE is calculated
   a. \( \text{RMSE}(S_3,T) = \sqrt{(0.2 - 0.1225)^2 + (0.8 - 1.0)^2} = 0.2144 \)
   b. At this point the system can either determine that this RMSE number is within tolerance levels of the target corpus, or it can equally decide that since the transform functions have been exhausted S₃ represents a best effort transformation for this problem.

Two aspects of this scenario are worth noting. First, the two marker-detectors M1 and M2 were considered with equal weight distributions. This does not have to be the case, the analyzer routine can assign alternative weightings to these markers, emphasizing or deemphasizing one of them. A new attribute could accompany each marker as a measure of relative importance. But since we have yet to find a good theory of indicating how important each marker should be, we continue to keep them all equally weighted.

Second, operators X1 and X2 were applied successively to the source text. The final S, (S₃) was therefore given by \( S_3 = X2(X1(S)) \). However, there are four possible orderings of these two operators: \( X1(S), X2(S), X1(X2(S)) \) and \( X2(X1(S)) \). What we did in this example is a sequential application of operators without any feedback considerations. But we could easily change the planning algorithm to perform exhaustive search (try every combination) or a greedy algorithm (apply the next step with the highest pay off), or a genetic algorithm with the distance as a fitness function or any other algorithm we wish.

This sub-problem is NP-complete and not really the main concern of this work. However we do allow for user specification of this algorithm.
2.3 Tools and resources

Abstract
A significant part of a system relying on heterogeneous methods is the strength and accuracy of the individual sub-systems. In our design we have abstracted all the heterogeneous methods into marker-detector functions and transform functions. In the background section we covered some theoretical and field work being done in various disciplines that could inspire creation of markers and transformers. In this section we examine various software tools and accessible online resources closely and discuss how we could use them in our system. This section exemplifies some of the most immediate work ahead of us in terms of software design.

2.3.1 AAAC corpus and JGAAP

Automatic Authorship Attribution (AAA) is perhaps the closest scientific effort to the style-based classification section of our project. Our own work [Kho05, Kho06] originated in that framework.

In 2004, Dr. Patrick Juola of Duquesne University hosted an Ad Hoc Authorship Attribution Competition (AAAC-04). The contest provided multiple corpora each with many documents (short essays, plays, fiction, etc) as well as a number of unlabeled texts. Over a dozen teams submitted classification programs to match the unlabeled texts to one of the corpora, or “none of the above”. The contest generated a high amount of interest in the community. The organizers declared the winners, and exposed the correct associations online. Since then, the AAAC corpus has become a popular research standard and many papers cite it. Although there has been talk of organizing another contest, the 04 corpus remains the most comprehensive and most recent of any such effort.

As of 2008, Dr. Juola announced work on an NSF funded project called Java Graphical Authorship Attribution Program (JGAAP) which shares many goals with the system we have in mind (except only in authorship attribution). The project was built on the work from AAAC and includes various algorithms and text features debuted there. The ongoing project takes some important steps in abstracting methods and algorithms for ease of testing and research purposes. Although the current version of the project is available to download, it is not freely modifiable.

Although the effort is specific to authorship attribution, the community has recently begun discussing more broad classifications beyond strict authorship. Multiple types of text from the same author as well as institutional authorship (i.e. New York Times, government reports, etc.) are also being considered. Cooperation with the Juola team and learning from lessons of JGAAP development would be very helpful to our effort. Adapting and incorporating authorship text features into style markers would be a valuable exercise.
2.3.2 Link Grammar Parser

Link Grammar Parser (LGP) is a popular linguistics and NLP tool, freely available. LGP is able to do parsing of English text and fairly accurate part-of-speech tagging. LGP is named after a theory in linguistics called “Link Grammar.” The parser has a number of features very attractive for our purposes. One is a “best effort” delivery in which even processing grammatically inaccurate sentences can produce some useful information. Another is a measure of complexity of sentences delivered by the tool.

We have used LGP in the past [Kho06] in a limited way as a marker to derive a measure of word constituent densities. However, there is much more that can be done with it. In addition to many possible markers indicating sentence type or complexity, LGP can be used to determine correctness of style transforms. For example, the transformation system could require that the source and transformed texts maintain the same level of complexity and pars-ability, reducing transformation errors.

2.3.3 Paraphrasing and summarization tools

There are many paraphrasing tools of various strengths available online. Some of these are done for entertainment purposes (i.e. Pirate Talk, Yoda Speak, etc.), but others are part of research efforts that involve beyond-the-surface linguistic manipulation. Paraphrasing of search words for the purposes of text retrieval and database interface systems is especially popular. Some of these methods could form the basis of transform operators in our system.

There exists a distinguished body of research in the area of automatic summarization. Research tools that can perform summarization on specialized corpora exist today. Many summarization tools rely on tagged input text, but some can operate on unstructured text as well. We believe pursuing summarization technologies and possible automated tagging which could enable them is a very fruitful area to explore and could yield new sets of transforms for us.

2.3.4 Word Net, ontology and word lists

Word Net, a freely accessible ontology of English words is a valuable resource and useful for both style markers and style operators. Word Net, returns “synsets” or set of synonyms for a given English word. Ontological distances between two words can also be extracted from Word Net. Information extracted from Word Net as a result of processing a set of words in a sentence could easily yield 10 or more distinct style makers. Of course the synsets themselves could be used for simple transformations by substitution, coupled with correct sense-of-word disambiguation.

Word Net is a large and comprehensive ontology of English words, but there is much other ontology available for specific domains. Given some fairly simple domain analysis, subject-specific ontology could be very useful in style transformations.
Lastly, word lists are valuable tools. The more defined groupings of words that are available for our system to access, the more ways we can develop to draw similarity and distinction between two corpora of text.

For example, a list of words and phrases associated with “radical feminism” could be very useful even for text that is not about radical feminism. First, if the text is about that subject, the “hit rate” of those words would be very high and detectable as a marker. Second, if the text is determined to be not about radical feminism, the word list could act as a constrain for transforms by requiring that the resulting transformed text have similarly low level of words associated with radical feminism. If the objective is to transform text to a corpus which does have a high hit rate on this word list, then the list could act as a positive constraint by giving preference to terms occurring in the list in case of Word Net-type substitutions where more than one synonym is available.

2.3.5 GNU diction and style

*Diction* and *style* are two older GNU utilities that have inspired many NLP researchers in the past. *Style* does surface analysis of text and provides some key raw characteristics of a document as well as popular readability scores. *Diction* is a word processing aid that makes a number of rewrite suggestions for common English misspellings, grammatical mistakes and overly complicated prose.

2.3.6 Readability measures and automatic scorers

In fields of writing and education, there are many readability indices available each giving different measure for clarity, sophistication or formality. In our recent work [Kho08] we examined several popular indices: Kincaid formula, ARI, Coleman-Liau, Flesch easy reading formula, Fog index, Lix formula and SMOG grading level. These formulas are often used in so-called automatic essay scorers such as the ones used by standardized testing services. The complete definitions for these indices are discussed in [Haa07]. Each can be a marker in itself, but in order to minimize variable dependences, we can deconstruct a superset of raw language elements that one or more of the indices consider in different proportions.

2.3.7 Lancaster University online course on stylistics

In 2005, professor Mick Short of Lancaster University created a new linguistics course called “Language and Style.” The course deals with stylistic analysis in literary texts. In 2006, all the course content, including sample passages and exercises were put online [Sho06]. The collection has serves as valuable tool for references and problems in stylistics.
3 Current status, work plan and evaluation

3.1 Current state of software

Our data from style processing work has been generated using a set of Linux scripts that perform the core of the functionality we hope to build for the prototype discussed in this document. The scripts are mainly written in bash scripts but LinkGrammar Parser and GNU tools awk, flex, diction and style are accessed as well. In the case of flex, c executables developed with flex are used for lexical analysis. There is no database interface at the moment as flat files are used to store vocabulary and interim statistics.

The current classifier is based on the work done in [Kho06] and uses naive Bayes. The transformer is made up of three major executables: getMarkers, analyze, and transform. Sample code for these is listed below at Appendices B,C and D respectively.

GetMarkers gathers statistics on a given text based on features specified in a text file. These features are currently only string based and do not contain any higher grammatically or statistically defined functions. Analyze runs getMarkers on target and current source and produces a distance (RMSE) between these texts.

The operator functions consist of known acronym expansion (based on text database), pronoun replacement (with first cited instance in paragraph), sentence splitting, paragraph splitting, and automatic text replacement based on GNU style suggestions library. Wherever possible care is taken to make sure that as much as possible the ambiguous situations are not operated on. For example, given multiple proper noun candidates for pronoun replacement in a given paragraph, the operation will not replace any pronouns.

The transform script executes a main loop in which it runs analyze then executes the next transform operator with the source text as parameter. Once done, analyze runs again, a comparison is made between “before” and “after” distance values and information is printed to the screen.

A very crude planner in transform uses a semi-greedy algorithm for selection of next operator: After the first iteration, if RMSE is found to be climbing, the next operator runs on the previous source file (as opposed to the latest one) instead, in effect negating the work of the last transform. Otherwise, the next transform runs on the latest source file which includes the work of the previous transform. At the moment the program halts when all enabled transforms have been executed once. Given the small number of transforms, we are generally not yet seeing sufficiently small RMSE values.

Using this current system we can show reduction in RMSE but we cannot yet show very small or zero values for RMSE, without trivially reducing the number and quality of detected style markers. Naturally, the preferred way to reduce RMSE is to run more and better style transform operators. There is no facility to easily switch between different algorithms and evaluation methods in analyze at the moment.
The scripts that we have developed mainly over the past year are inadequate for this project. Due to the interpreted nature of *bash*, there are terrible speed performance problems that make using larger corpora impractical for what we have in mind. In addition, the primitive GNU tools and command line interface has caused the project to reach a point where further development under this environment will be very inefficient. This is why we find it necessary at this point to begin building a new prototype system using the more modern methods, compiled code and more robust tools keeping in mind the goals of heterogeneous methods and modularity.
3.2 Project plan and timeline

Abstract
We have constructed the following timeline for accomplishing the major goals of this project in 12-20 calendar months starting January of 2009. During this time, we aim to build a more robust and powerful system, collect and encode as many style markers and transform functions as we can, consult human experts as to the competence of the transformations and continue our publishing schedule along the way. Lastly, we would like to develop a real-life application prototype, such as those discussed in section 1.2 which would interact with the underlying system. These are the major milestones that we have included in the project plan below.

Phases of the project (beginning January 2009) and the milestones for each phase:

- Phase I: Vertical prototype (3-5 months)
  - Build the skeleton of the prototype system using the new programming tools and web environment. The goal is to get it to the level of the scripting based system we currently have. Very few markers and very few transformers will be implemented other than some token ones to demonstrate the feasibility and modularity.
  - Install and facilitate database communication with the web based system.
  - Gather appropriate corpora for testing.

- Phase II: Build up collection of markers and transforms (3-5 months)
  - Include more markers and transformers using researched resources and techniques. Consult more linguistic experts and tools in order to develop more robust and more generic natural language manipulation tools. Markers should be empirically tested in strict classification exercise to determine their relative values.
  - Work on planning algorithm for transformer selection, as well as statistically based markers (i.e marker functions that are based on specific text features and will be different for different corpora).

- Phase III: Evaluator, comparator and planner (3-5 months)
  - Work on automatic normalization and weight adjustment for the marker vector based on reinforcement learning.
  - Work on alternative and interchangeable distance metrics between texts (as opposed to strict style matrix RMSE).
  - Work on abstracting the planning function and allowing user control over its operations.

- Phase IV: Transform granularity and constraint satisfaction (3-5 months)
  - Develop a granularity operator for transforms or degrees of freedom (i.e. even though applicable to many sections of a text, the transform functions should manipulate very few sections of the text at one time) and a measure of granularity should be specified and utilized by the system.
o Improve transformation performance by adding more constraints for basic meaning transference (i.e. grammatical correctness and measurable meaning orthodoxy).

o Test transformation with more corpora and validate by using best classification algorithms.

o Develop workable prototypes for one or more real-world applications.
3.3 Evaluation

Abstract
The evaluation process for any classification system is trivial given labeled corpora. Evaluation of transformation function of the system can be divided into machine and human verification. Machine evaluation uses classification to evaluate the success of a stylistically transformed document. Human evaluation depends on area experts making judgments about degree and effectiveness of the transformation toward a target style.

3.3.1 Classification
We simply use well-known machine learning techniques of dividing the corpora into training, test and validation. Our system would then train with a labeled training set and run trials with unlabeled test and validation sets. An objective success rate can be calculated and used for overall evaluation of the system.

3.3.2 Machine and human verification for transformation
The evaluation of the system response is divided into two categories: machine and human evaluation. Human evaluation mainly applies to transformation as classification has objective logical criteria that can be verified by automated processes. However, part of the statement we are making with this project is that transformation also has some objective logical criteria that can be verified automatically.

3.3.2.1 Machine verification of transformation
What is the ultimate test of transformation if not having the transformed text be automatically classified as belonging to the target corpus? In fact there is already built-in verification inside our system, the distance formula between source and target is defined in the same terms as the classifier would use to classify. However, as an objective scientific test that classification is not adequate. We will apply other classification algorithms to verify that the transformed text, indeed, can be considered as part of the target corpus.

3.3.2.2 Human evaluation
We have made the bold claim that our transformation system can change the style of a written text. This cannot be measured entirely by machines and needs human expert approval. The reason is simple: we can easily imagine a source document be successfully
and confidently machine classified, yet when read by a human, the text sounds and feels nothing like the corpus it is transforming to.

Does this discrepancy between machine and human experts suggest a failure of transformation or a failure of classification? We have argued that automatic style transformation is only as good as the best classification and that failure of transformation is also an indication of under-specification of the target corpus style.

Given constraints and impracticalities of this project we almost certainly won’t be able to conduct any large scale human studies. A single human subject “test” would take impractically long as the subject must familiarize himself/herself with the style of the target corpus before attempting to decide if the source text is written in the same style.

One seemingly good solution might be to simply use well-established authorial figures as the target corpus hoping that human subject would already know the said style and can evaluate the transformed text without having to read the entire corpus. There are however, problems with this approach: First, what this necessarily means is that human subject is reacting to a perception of the target style which certainly brings in external elements from the author’s brain and experiences into the picture, making it difficult to determine conformism to a style that may be objectively different than the one the user has perceived it in the past. Equally problematic would be if the source text has cartoonishly exaggerated clues about the target style and that would increase chances of these clues be recognized by a human subject.

For these reasons, we see value in periodic evaluations by small numbers of human experts who can be conscious of the pitfalls. We propose selecting two faculty members with backgrounds in writing or humanities evaluating several transformation problems at the end of each phase. These experts would provide feedback as to what could be done to make the system stronger. The last evaluation calls for more experts, perhaps as many as four. The interim results would be incorporated into the system in form of better style specification or better transformation algorithms. By the end of the project, we would strive to have 10-12 sets of feedback evaluations from the human expert subjects.
4 Summary

4.1 The proposal

The subject we are concerned with is the nature of writing styles. We would like to understand writing styles through having effective automated methods to classify and transform them. We propose to develop a system that performs automatic writing style classification and transformation. We further propose that this system be modular, extensible, web-accessible and make use of heterogeneous processes to accomplish its goals. Using this system, we hope to derive increasingly sophisticated understanding of natural language writing styles.

4.2 Background

The background section of this document discusses various forms of human-agent enabled style processing in various humanities disciplines. We also survey work in text classification, authorship attribution and computational stylistics, as well as natural language generation systems and digital writing assistants.

4.3 Approach

Our approach can be broadly described by the “classification-transformation” loop. This is a process whereby an inadequate result of a style transformation operation drives a more precise specification used in the classification process. And a more precise classification process in turn drives a more sophisticated transformation process. We hypothesize that if we keep continuing in this loop, at some point the machine transformation results will yield samples nearly indistinguishable from original work, even as judged by a human expert.

4.4 Plan

Our plan is a systematic software engineering of the classification-transformation system. Beginning with the requirements, we plan to develop a vertical prototype or skeleton of the system, and then continue to augment it by plugging in more and more sophisticated style marker detector, and style transform functions.

4.5 Targeted contribution

First, we hope to build an automated example driven automatic stylistic transformation system which we believe would be the first of its kind. Second, we would like to advance research in the area of classic style processing in the humanities by providing a user-friendly system for researchers to derive a set of relevant stylistic features from a corpus. Third, we would like to help the computational linguistics and natural language processing
communities by providing a modular system that can act as a test bed for various style markers and style transform functions.
5 GLOSSARY

Classification. Process of matching a document to a known / profiled corpus.

Corpus. A collection of documents determined to be stylistically related.

Lexeme. A set of inflected forms taken by a single word.

Marker / style marker. A particular feature of the text determined to be style related.

Marker-detector function. A function that returns a value indicative of the level of presence of a certain marker in text.

Operator / transformation operator. An atomic operation performed on text resulting in different text. Operators operate on words, sentences or paragraphs.

Profiling. Process of extracting all marker related statistics from a corpus.

Source. Term generally given to a document whose style is to be transformed.

String. A set of characters followed by an end of string marker.

Style. We contextually define style as a unique and consistent set of options exercised while writing a corpus. We attempt to model style of a text by its style signature.

Style signature. A particular unique combination of style markers observable in a corpus or a source text.

Target. Term generally used for a corpus whose style is being profiled in a transformation.

Transform. A transform is made of up of one or more semantically related transformation operators.

Transformation. Process of altering the style signature of text.
6 Bibliography


[Fis81] Fish, Stanley, “What is stylistics and why are they saying such terrible things about it”, in Essays in Modern Stylistics, edited by DC Freeman, Routledge, 1981, pp 53-66.


7 Appendices

7.1 getMarkers.bash script

#!/bin/bash

#getMarkers.bash: This program calculates marker statistics from a given text file.

echo "#DOCUMENT:"
totalWords=`cat $1 | wc -w`;
echo "# - words per: "$totalWords;  
echo "# - characters per: `cat $1 | wc -m`
#echo `cat $1 | wc -wm`
rm -f $1.wm $1.uniquewords $1.wordstats

for paragraph in `cat $1 | awk -F \"\n\" '{if ($1 != "") print $1}' | tr ' ' '_'`
do
    echo -n `echo -n $paragraph | tr '_' ' ' | wc -wm` >> $1.wm
    echo -n " " >> $1.wm
    echo -n `echo -n $paragraph | awk -F '?' '{print NF-1}'" " >> $1.wm
    echo -n `echo -n $paragraph | awk -F '!' '{print NF-1}'" " >> $1.wm
    echo -n `echo -n $paragraph | awk -F ':' '{print NF-1}'" " >> $1.wm
    echo `echo $paragraph | tr '.' '@' | tr '!' '@' | tr ':' '@' | tr '_' '
' | grep -c '@'` >> $1.wm
done

n=`wc -l $1.wm | awk '{print $1}'`

sum=0;
for stats in `cat $1.wm | awk '{print $1}'`
do
    sum=`expr $stats + $sum`
done
echo " - avg words per: "`echo $sum $n | awk '{print $1/$2}'`

sum=0;
for stats in `cat $1.wm | awk '{print $2}'`
do
    sum=`expr $stats + $sum`
done
echo " - avg characters per: "`echo $sum $n | awk '{print $1/$2}'`

sum=0;
for stats in `cat $1.wm | awk '{print $7}'`
do

sum=`expr $stats + $sum`; done
echo " - avg sentences per: "`echo $sum $n | awk '{print $1/$2}'`; echo "#WORDS:

echo `cat $1 | tr '.' '\' | tr ',' '\' | tr ':' '\' | tr '?' '\' | tr '!' '\' > temp
cat temp | tr ' ' '\' | grep [a-zA-Z] | tr a-z A-Z | sort > $1.allwords uniq $1.allwords > $1.uniquewords

# gather statistics in this format: word frequency_in_doc length freq_in_dict
for w in `cat $1.uniquewords`
do echo "$w" `grep -icw $w $1.allwords" "`echo $w | awk '{print length($1)}'" "`look $w | wc -l` >> $1.wordstats; done

n=`wc -l $1.wordstats | awk '{print $1}'`
sum=0;
for stats in `cat $1.wordstats | awk '{print $2}'`
do sum=`expr $stats + $sum`; done
echo " - avg relative frequency per: "`echo $sum $n | awk '{print ($1/$2)/$totalWords}'`; sum=0;
for stats in `cat $1.wordstats | awk '{print $3}'`
do sum=`expr $stats + $sum`; done
echo " - avg characters per: "`echo $sum $n | awk '{print $1/$2}'`; sum=0;
for stats in `cat $1.wordstats | awk '{print $4}'`
do if [ "$stats" -gt "0" ] then sum=`expr 1 + $sum`; fi done
echo "#VOCABULARY:"

echo " - linux dictionary hit rate: "`echo $sum $n | awk '{print $1/$2}'`;
### 7.2 Sample output from a demonstration program

```
[root@ayandeh st]# ./transform.bash turing.txt source
################################ transform script started
################################ Analyzing pristine source and target

turing.txt Marker Stats
#DOCUMENT:
# - words per: 2921
# - characters per: 17439
# - paragraphs per: 32
#PARAGRAPH:
- avg words per: 91.2812
- avg characters per: 543
- avg sentences per: 4.6875
#WORDS:
- avg relative frequency per: 0.00106724
- avg characters per: 6.75667
#VOCABULARY:
- linux dictionary hit rate: 0.979723

source Marker stats
#DOCUMENT:
# - words per: 211
# - characters per: 1465
# - paragraphs per: 3
#PARAGRAPH:
- avg words per: 70.3333
- avg characters per: 486.667
- avg sentences per: 3.66667
#WORDS:
- avg relative frequency per: 0.00729927
- avg characters per: 6.63504
#VOCABULARY:
- linux dictionary hit rate: 0.970803

RMS error: 2.38781
################################ Running Transform 1 on source

turing.txt Marker Stats
#DOCUMENT:
# - words per: 2921
# - characters per: 17439
# - paragraphs per: 32
#PARAGRAPH:
- avg words per: 91.2812
- avg characters per: 543
- avg sentences per: 4.6875
#WORDS:
- avg relative frequency per: 0.00106724
- avg characters per: 6.75667
#VOCABULARY:
- linux dictionary hit rate: 0.979723

source.t1 Marker stats
```
source.t3 Marker stats

#DOCUMENT:
# - words per: 214
# - characters per: 1478
# - paragraphs per: 2

#PARAGRAPH:
- avg words per: 107
- avg characters per: 737
- avg sentences per: 5.5

#WORDS:
- avg relative frequency per: 0.00714286
- avg characters per: 6.41429

#VOCABULARY:
- linux dictionary hit rate: 0.992857

RMS error: 2.33089