Using Rhetorical Figures and Shallow Attributes as a Metric of Intent in Text

by

Claus W. Strommer

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2011

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

ABSTRACT

In this thesis we propose a novel metric of document intent evaluation based on the detection and classification of rhetorical figure. In doing so we dispel the notion that rhetoric lacks the structure and consistency necessary to be relevant to computational linguistics. We show how the combination of document attributes available through shallow parsing and rules extracted from the definitions of rhetorical figures produce a metric which can be used to reliably classify the intent of texts. This metric works equally well on entire documents as on portions of a document.

ACKNOWLEDGEMENTS

It is difficult to overstate my gratitude to my Ph.D. supervisor, Chrysanne DiMarco. Through enthusiasm, inspiration, and encouragement she provided me with guidance without which I would have felt lost in my work.

I would like to thank the people who directly guided my thesis, especially Randy Harris for introducing me to the academic study of rhetoric, Charlie Clarke for guiding me through the pragmatics of applied corpus research, and Robin Cohen for her unerring eye for detail in thesis writing.

I would further like to express my gratitude to the many people whose input and contributions shaped the body of my research, primarily Jeanne Fahnestock, Vessela Valiavitcharska, Olga Vechtomova, Nike Abbott, Ashley R. Kelly, Robyn Roopchan and George Ross.

I am grateful to the secretaries and directors in my department for assisting me in many different ways and always supporting me and encouraging me to carry on. Margaret Towell and Yuying Li deserve special mention.

Lastly, and most importantly, I wish to thank my family: My wife, Diane Principato, my sister Sonia Bynum, and my mother, Carmen Berkowitz, and my late stepfather, Walter O. Spitzer. Without their love and support I would not have been able to attain this achievement. To them I dedicate this thesis, and to them go the laurels of my success.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Statement of Intent

In this thesis we propose a novel method of metering natural language text based on its rhetorical attributes. Unlike previous approaches which focus on deep analysis of the text, our method is capable of generating significant and consistent evaluations based on shallow attributes of the text. We achieve this by using information gathered from rhetorical figures to understand how the shallow attributes of text can be combined to form “emergent” complex attributes – attributes whose complex structure is composed of shallow attributes combined in a specific pattern – upon which our metrics are based. In this process we recognise the study of rhetoric and rhetorical figures as a field which is rich in data but whose seeming lack of structure has detracted scholars in natural language processing from mining its knowledge. In this thesis we dispel this notion, showing how a meticulous examination and careful interpretation of rhetorical figures can yield information about documents that is useful not only to computational linguistics but also to any field working with natural language text such as document classification and information retrieval. We furthermore generate specific values of saliency based on detection and classification of rhetorical figures in documents which can be applied towards a metric that in turn can be used to effectively and efficiently evaluate author intent at both the macro- and micro-document level. This flexibility of application distinguishes our method from other similar metrics based on shallow attributes which do not apply to micro-document level evaluation.

1.2 Chapter Summaries

We begin in Chapter 2 by reviewing the current state of natural language processing, natural language generation, and document evaluation. We use this review to show that there is an open field in document evaluation, how our research fits that field, and what additional knowledge from the field of rhetoric benefits our research. Most importantly we determine what it is that we want to measure with our proposed metric. We then proceed in Chapter 3 to structure the depth of our research. We choose epanaphora - the repetition of words at the beginning of sentences, phrases, or lines of text - as the rhetorical figure focus for our study. As a figure of syntax (schemes), epanaphora is an ideal candidate for this research. Because it is based on repetition it can be reliably detected without using deep analysis of text. We distinguish between intentional and accidental epanaphora, how they differentiate, and how this distinction is relevant to our metric for natural language evaluation. We determine what shallow attributes we use for epanaphora detection, how we use these attributes, and we demonstrate that the use of these shallow attributes returns a comprehensive list of candidates for epanaphora. Chapter 4 discusses the refinement of the rhetorical figure detection process. We propose additional shallow attributes which we use to distinguish between intentional and accidental epanaphora. Since the attributes in Chapter 3 are used to perform comprehensive epanaphora detection, we apply these new shallow solely towards recording the new features of existing epanaphora, which are then used for classification between intentional and accidental epanaphora. We conclude this chapter by redesigning the detection- and attribute recording algorithms to efficiently accommodate the new shallow attributes determined in this chapter. We then proceed to Chapter 5, which discusses the main study of this thesis: the automatic classification of rhetorical figures. We proceed through the tasks of using expert annotators to create an annotated corpus of rhetorical figures to train automatic classifiers, followed by the initial training and evaluation of different models of automatic classification. After that we perform a refinement of the classification process through the application of a third generation of shallow attributes as constraints. Lastly we apply the gained knowledge to a new corpus and evaluate the results.

Chapter 2

Background

2.1 Area of Research

As a field under the large umbrella of artificial intelligence, computational linguistics (also known as natural language processing) is the broad area of research whose focus is to understand and generate the interaction between computers and human (natural) languages [77]. These constructs are most commonly in the form of text, but research on the understanding of spoken language also exists.

Natural Language Generation (NLG) is a sub-area of Natural Language Processing (NLP), and focuses on producing human-readable text from information sourced from computer-understandable representations. It is a subtopic to the fields of Artificial Intelligence (AI) and a sub-field of computational linguistics. NLG has a close link to Natural Language Understanding (NLU), in that they are effectively complementary methods. Whereas NLU focuses on interpreting human language –

which is often ambiguous and potentially malformed – into unambiguous computer representations, NLG's efforts are put into taking the computer representations and producing human-readable output that meets particular reader criteria. The primary driving goal behind NLG is best described by Reiter and Dale [121], who write: ”Natural language generation is best characterised as a process of CHOICE: Given the different means that are available to achieve some desired end, which should be used?”

In this chapter we examine the diverse fields of natural language processing and follow the process of discovery used to anchor the research of this thesis. We begin with an overview of NLP and the diverse ecosystem of metrics and classification algorithms. During this overview we draw up the skeleton of our thesis and identify the possible strengths of our approach.

2.1.1 Brief History of Natural Language Generation

Early work in natural language generation started in the 1950s and 1960s and focused on machine translation and generation grammars for generating well-formed sentences. However, work on actual generation of natural language content (as opposed to mapping from predetermined content) did not begin until the mid-1970s, with work by Goldman and Davey [66] [98]. The most important contribution of these early efforts is the establishment of a distinction between natural language understanding and natural language generation. During the early 1980s the efforts by McKeown and Appelt [99] [3] helped shape the direction of natural language generation for years to come. The most important trend developed in by this decade was the advent of specialised systems that focused on one particular task of natural language generation, as opposed to the monolithic systems that had been prevalent until then. The 1990s saw a large number of new practical applications of NLG technology [121], many owing their success to new technologies that had previously made their deployment impossible or impractical. In the field of natural language understanding, more powerful computer systems made it possible to introduce statistical analysis of text as a viable option. Since the late 1990s there has been a rapid growth of work on multi-agent and distributed systems. These efforts are starting to be incorporated into natural language generation through works such as that of Hervas and Gervas [71] and ongoing work based on ideas first proposed for the HealthDoc Sentence Planner [146].

2.1.2 Natural Language Generation Systems

Classic Model

Early natural language generation systems had two stages: Document planning, during which the content and overall layout of the text is generated, and surface realization, which took the document plan and performed the final tasks of the generation process, such as syntax and morphology among others [121]. However, a number of tasks such as aggregation [43][71], coreference [3][42], and lexical choice involve knowledge that is relevant both to document planning and surface realization. Thus a third step was conceived – Microplanning, also known as sentence planning. The introduction of this step has helped streamline the natural language generation process by allowing the document planner to focus solely on the coarse-grained aspects of generation, and taking the decision-making steps out of the surface realization task.

Aggregation (also called ellipsis or coordination in linguistics [43]) and coreference share a large number of characteristics in that they both perform the task of reducing text monotony by eliminating repetition. They differ in what kind of repetition is eliminated, and how this is achieved. A key element in these steps is that even though repetition is eliminated, the aggregated/coreferenced text does not lose any explicit information.

Lexicalization is the part of microplanning that reaches the furthest into both document planning and surface realization. The task of lexicalization is to take the document plan's skeleton (or template) and introduce the appropriate terms (words) which are then taken by the surface realizer and converted into the final sentences.

The work of near-synonymy falls close to lexicalization in that it involves the selection of appropriate words. However, unlike lexicalization, near-synonymy is less involved with the document plan, because its main goal is to eliminate single-term repetition by introducing synonyms and near-synonyms.

Agent-Based Systems

Monolithic systems in computer science are becoming less common with the advent of accessible distributed computing resources. While monolithic models provide the best solution for systems that reside on one machine, they are limited to the resources of that machine. Distributed systems show better overall performance due to their scalability, as long as the task given does not depend on low-latency communication.

As a result, one can think of agent-modelled applications to be the embodiment of high-latency distributed systems. It is expected of agent-based systems to separate the given tasks as much as possible, thus minimising the amount of information that needs to be communicated between agents.

In programmable agent-based systems [35], the task of assigning work to different agents falls upon the *facilitator*. This facilitator can be seen as being the gateway agent through which requests are submitted. The facilitator will then decide which agents can perform the requested tasks and which of those agents are available. If the requested query needs to be performed by multiple agents, the facilitator also handles the coordination of those agents. This agent-facilitator model is more flexible than parallel-programming systems, because it does not rely on 100% uptime of each agent. It is possible to dynamically add or remove agents as well, thus easily changing the way the system handles queries.

An example of an agent-based text generation system is showcased by Pereira, Hervás and Gervás [112], with different agents handling the lexical database, lookup of concepts related to a lexeme, structure alignment, and query generation.

In our thesis we adopt a hybrid model combining centralised NLP and ideas from agent-based systems. In particular we strive to make our system modular by maintaining low coupling between each of the 'agents', i.e. the stages of the software. We also maintain high cohesion within those stages. Through this approach we eliminate the need of a low-latency network between each 'agent' but still allow them to act independently from each other.

Content Adjustment

So far, our discussion of computational linguistics has focused on generation of new documents. However, a significant portion of the research in natural language generation can be placed under the umbrella of text-to-text transformation. Among the more noteworthy fields of research in this area are translation, summarisation, and generation by selection and repair [49].

The goal of machine translation is to interpret text in one language and reproduce it, usually in another language. This process involves both natural language understanding and natural language generation, and as such has to deal with difficulties from both fields. In its simplest form, machine translation is done on a word-by-word basis. This process is known as token substitution. This simple approach has been shown to produce relatively good results when the source and target language have similar syntactic and grammatical constructs. However issues can still arise with compound words and words which are a single lexical entity in one language but not the other. Brown et. al. [23] give the English-French translation *to_go→aller* as an example of such issues: "While our model allows many target words to come from the same source word, it does not allow several source words to work together to produce a single target word".

The task of extracting content from an information source and displaying it in a condensed form is called summarisation. The types of summaries vary based on the user (generic or personalised) and the content of the summary (extract vs. abstract) [94]. Automatic summarisation plays an increasingly important role in today's society. The ever-increasing volume of online documents translates into an inability by users to keep up with information they need to know and/or be aware of. Summarisation of online documents differs from traditional goals of summarisation in that much of the information available online is duplicate among several sources. Generation of summaries that is consistent with all sources for which the content overlaps is important for services such as news aggregators and search engines [131]. Mobile computing introduced yet another paradigm shift in summarisation due to the limited physical resources of mobile devices and the consideration that services are no longer dealing with stationary users, meaning that the information relevant to an access terminal changes with its location.

Translation and summarisation systems generally process their input only once. However, with personalised online services such as online portals and content management systems becoming the norm, there is an increased demand for systems of source reuse. Key to source reuse is the idea that it is possible to edit individual portions of the text once. These sections are internally consistent, thus removing the need for the generation system to verify their correctness and instead allowing it to focus on assembling them into the final, full document. There is a limit, however, to how much editing can be removed from post-assembly. The difficulty of pre-editing increases significantly with each additional customisation option, specially when a comprehensive set of variations is required. Ensuring that every change in the source remains consistent with each combination becomes a NP-hard problem. An alternative was proposed by Wanner and Hovy [146]: Pre-editing is limited to assuring internal consistency for each individual source segment inside the 'Master Document'. After assembly, a second editing step 'repairs' transitions between segments to guarantee coherence. This approach of separating content *generation* from content *correction* significantly lightens the burden of content generation, and has been used in commercial applications of text reuse involving automated generation of tailored web-page and print-based health education materials [49].

Even though effective techniques for summarisation have been established already, the choice of using one of these techniques over another is still driven by human selection. The reason for the need of human input is the lack of a standardised metric for evaluating the effectiveness of each summarisation technique against different human demographics. We find that this is a common problem in natural language processing applications. The root of this problem is the inherent subjectivity of evaluation from the perspective of a human demographic.

2.1.3 Statement of Problems

All of the systems described in Section 2.1.2 share the common trait of having output that is constructed from pieces of input text that already are internally coherent, and in many cases pre-fabricated. As a result, many evaluation methods for natural language generation do not apply to them. Text-to-text transformation systems that put a focus on recycling input material do not generally affect the traits of generated text that classic methods of evaluation examine.

We propose that it is possible to devise a metric for evaluating the effectiveness of natural language generation frameworks by how well they *preserve* the local coherence of the input. This method is superior to classic systems of measurement which focus on generating a single metric for an entire document. Instead of treating the entire document as an immutable and homogeneous input our approach is sensitive to internal variations and changes between the inputs and outputs of text generation systems. However, unlike token-based models for shallow localised metrics such as Markov chains, we maintaining a sufficiently coarse granularity: By using entities that are larger than simple words we prevent out system from being overly sensitive to minor lexical changes.

We further propose that the same metric used for evaluation can also be used in document classification. A measurement that is a composite of various document attributes would be capable of exposing intra- and inter-document relations which are not evident otherwise.

2.2 Related Fields of Document Evaluation

In Section 2.1.3 we described one of the open problems in natural language processing as the need to produce a metric for the evaluation of NLP systems as well as for document classification. In particular this metric should produce more than a single value per document while still being generic enough to be resilient to fluctuations at the shallow text level. We proposed local coherence as one of the possible ways in which this metric could be quantified.

In this Section we will examine the existing systems of metrics in computational linguistics, how their achievements converge towards or diverge away from our domain, and how these systems of metrics can contribute to this thesis.

2.2.1 Readability Indices

Readability is the quality of a text that makes it more compelling and easier to comprehend than others. It is not the same as legibility. The latter focuses on typeface, calligraphy, and layout, among others, and is generally studied under the subject of typography [20] [31] [142] [108]. Readability is purely dependent on writing style, and has been attributed to clarity [69]. Readability affects rate of understanding, reading speed, and reader interest [41].

Readability Guides and Formulas

There are numerous guidelines for English composers to make their text more readable. For example, standard rules for documentation writing include, among others, the use of short, simple words, avoiding domain-specific terms, being culture-and-gender-neutral, and avoiding complex sentences [134]. The aim of these guides is to improve the readability of text. Their success is usually measured in the form of an index based on word complexity and sentence length. Numerous formulas have been developed since the 1920s that attempt to predict the difficulty level of texts, with wide success in areas such as journalism, law, health care, and military documentation [51]. Despite their success and popularity (or perhaps because of it), there have been many critics of readability formulas. Much of this criticism lies in the perceived shortcomings of these formulas, namely that they ignore components such as cohesion, number of items to remember, rhetorical structure, etc. [11]. What these critics fail to realize is that while not perfect, readability indices provide a simple yet surprisingly accurate and objective method of predicting the reading level required by a human to comprehend a text. Most importantly, they do not require a perfect parse to achieve satisfactory accuracy, unlike more complex methods of quantifying readability. Some of the most popular indices of readability are the Flesch-Kincaid Grade Level formula [59], the Gunning Fog test [52] [68], and the Coleman-Liau index. The scores of the Flesch-Kincaid test and the Coleman-Liau index give the grade level required for a reader to understand a test. The Gunning Fog test calculates the number of years of formal education needed to understand a text.

The greatest contribution of readability formulas to this thesis are their extensive research on syntax-independent characteristics of documents. They show that it is possible to generate meaningful and useful metrics without deep analysis of documents and provide a solid background on the types of patterns that can be of interest to algorithms following a similar approach.

Limitations in Application

Unfortunately, readability indices are not very useful beyond their intended purpose. Firstly, there is a dissonance between the goal of readability indices (to help authors maintain their documents at a certain required reading level for the user) and the necessity of authors to maintain a particular sentence structure. In particular, the consensus among the proponents for the use of readability indices appears to be that a lower required reading level is better for any document. We contend, however, that it is just as important to maintain a minimum level of reading difficulty to maintain reader interest. Structuring a document to retain reader interest is usually achieved in natural language generation during the microplanning phase. The process of microplanning includes techniques like aggregation, referring expression generation, and coreference resolution [43] [66] [71] [98] [99] [3], most of which generally increase the complexity of the generated text according to readability indices. Readability indices indicate that text is more comprehensible if the word per sentence ratio (or syllables per sentence, or words per paragraph, depending on the particular metric used) is lowered, and that a too-high ratio will deter less skilled readers. Aggregation, on the other hand, is based on the principle that small successive sentences can be fused in order to reduce tedium due to repetition.

The second problem with readability indices is that they generate metrics at the macro-document level, whereas we desire a more fine-grained set of results at the intra-document level, that is, between the entities which constitute each document. For example, a change in the syntactic structure of a sentence has the potential to destroy some of the meta-information encoded in it, while not significantly altering the readability scores of the text. We wish for our system to be aware of syntax-level changes.

Lastly, while the use of readability indices in obtaining objective information about a natural text is undisputed, their feature-agnostic nature itself means that they are unfit for measuring less general attributes like affect, attitude, and semantic similarity.

2.2.2 Stylometrics

In Section 2.2.1 it was discussed why the statistical analysis of token counts in text (where a token can be a letter, syllable, word, or other unit that is not subject to interpretation and can be counted) is not useful as a method of quantifying change in natural text, nor is it capable of providing any semantic information at the phrase or sentence level. In particular, we need to find a way of analysing elements of the text that alter or enhance the meta-information provided by said text. In the context of text element analysis, meta-information refers to anything that can not be inferred through direct extraction of the explicit meaning of tokens, as done for example during a dictionary look-up. We are particularly interested in the stylistic features of natural text, since alteration of style is one of the key functions of document tailoring and, by association, narrative tailoring.

Whereas readability indices focus on counting the elements of a text that alter the grade level required to read a text, the practice of stylometrics is more interested in generating statistics based on "unit(s) of counting which (translate) accurately the 'style' of the text" [74]. Style can thus be defined as the set of elements that can be used to identify the writings of an author. It is for these reasons that stylometrics is also known (and more commonly practised) as *authorship attribution*.

Style Markers

There are two parts to authorship attribution. The first task is to extract the style markers that may be used to identify an author. In this endeavour, it is a common misconception of the uninitiated that it is sufficient to look at the rare or technical words (also known as peripheral or marginal words) used in a document. However, other markers such as context-free 'filler' (function) words [4] [27] [75] [82] [150] and punctuation symbols [4]. Part-of-speech tags have also been used in authorship attribution [5] [88] [130] [132] [133] [151], indicating that the same elements used in readability indices. There are similarities between the methods used on the above markers and readability indices, particularly with function-word markers. However, the major difference consists in readability indices being agnostic to the type of word used in their statistics.

Comparison Mechanisms

The second task to effective authorship attribution is the selection of a proper comparison mechanism. Particularly measurable features of the markers are word clustering, entropy, and the group of words encountered only once in a document, hapax legomena [14]. This latter category of text markers is interesting in that it may be used in studying the richness and expansion of an author's vocabulary over the course of their works. When it comes to methods used for analysing these features, statistical analysis comes out as a popular one, involving techniques such as linear discriminant analysis [4] [130] [133] and component analysis [5] [27] [75]. Another method is to approach the problem of authorship attribution as a document classification issue, in which case machine learning (Bayesian Networks, Support Vector Machines (SVMs)) is used [48] [87].

Relevance

There exist limitations to how authorship attribution can be applied to study semantic properties of natural language text. First, there are the constraints necessary for authorship attribution to be effective. These are the need for a limited and well-defined set of putative authors, a lower bound on the length of the text to be analysed, and the comparison texts should be corresponding in size or degree to the disputed document [6]. The second and third constraints are a disheartening quality in that they put hard boundaries on the flexibility of the authorship attribution methods. The metric system in this thesis is intended to be a general-purpose method of examination. By having to exclude documents below a certain length we would be discarding a significant portion of the applications for this metric. Evaluation of summarisation would definitively be out of the question, as the entire purpose of that practice is to produce a significantly shorter document. It is also doubtful, in the context of examination of text-to-text transformation systems, that there will be a significant divergence between the count of quantitative markers used in stylometrics between pre- and post-transformation documents. Generation by selection and repair [49] in particular would have the least variation, since most (if not all) of the repair is done at the grammatical level, and then only centered around the transitions between selected snippets of text. Lastly, not being able to perform analysis at intra-document level as defined in Section 2.2.1 means that atomic changes like those we expect to see in generation by selection and repair can not be easily identified due to the global nature of the measurement.

Despite the above reasons, the work in this thesis can still benefit from the study of authorship attribution. The largest contribution stylometrics is the multitude of options available for counting markers, in particular when it comes to identifying unique elements of style. As with the attributes of documents used in readability indices, these style markers are commonly independent of the syntactic structure of documents.

2.2.3 Local Coherence

Natural language generation systems need to not only ensure that their output is syntactically and grammatically correct, but also coherent. The study on coherence differentiates between local coherence and global coherence. The latter depends on the intentional structure of the document, that is, the structure that emerges from the document planning element of classic natural language generation. Local coherence, in contrast, is related to the coherence among utterances within each discourse segment of a document, e.g. sentence-to-sentence transitions and the perceived continuity between their constituents [67]. This perceived continuity is the aspect which is most affected by the types of generation frameworks in which we are most interested. In particular, once can consider the repair aspect of generation by selection and repair as being focused on restoring local coherence when there is a shear caused by selected adjacent segments of text being generated independently of each other. Similarly, summarisation preserves the general coherence of the document by keeping skeleton structure of the input, but local coherence suffers on account of sentences or segments of sentences being removed.

Local coherence, has been the focus of extensive studies in computational linguistics [67] [96] [10] [9] as well as psycholinguistics [100]. The approach to local coherence in which we are most interested is the entity-based model, which is supported by centering theory [139] and alternative approaches [135]. Entity-based study of local coherence examines the transitions between the given entities (usually words, phrases, or sentences) as sequences. These sequences are used to expose the quality of coherence of a generated document, and have been demonstrated to be useful in evaluating the coherence of summaries and assessing the readability of documents [9].

Salience

Salience, in regard to the study of local coherence, is the discriminating factor used to determine the degree of coherence among entities. The basic idea is that there is a very strong relation between salience and the degree in which referring expressions and pronominalisation occur [145]. These relations have been further extended to include topicality, predictability, and cognitive accessibility, and quantification has been adjusted from a binary representation to a hierarchical representation [9]. Common among all representations is that the way salience is viewed is by how entities are introduced and discussed.

The theory linking salience and coherence is simple: Coherence is seen as a strong continuity between discourse entities, as opposed to sudden and frequent switches between topics. More salient features are strong indicators of coherence, since salience is seen as a feature of topic continuity. By formalising the transitions between salient features it is thus possible to quantify the local coherence of a document.

Relevance

The study of local coherence is the area of research that most closely resembles the features of measurement that are of interest to us in this thesis. The most important contribution is the consideration of local transitions between discourse entities, the entity granularity itself. There are a number of points, however, where we believe that an improvement is possible. First is the selection of the type of discourse entity. Local coherence and centering work with syntactic entities at large. We believe that it is important to attempt to separate syntactic constructs from our metric, so an alternative choice of entity is necessary. Our key criteria are the aforementioned disassociation from syntactical and grammatical resources, and the preservation of granularity similar to the one used in local coherence studies. Anything larger would not be able to efficiently discern transitions between discourse entities, while anything smaller would regress into dependence on syntax.

Given that we will be relying on different entities as markers for our metric, we believe it is prudent to examine alternative definitions of salience. Research on local coherence has, as to date, relied on counting methods that closely resemble those used in readability indexes. We intend to introduce additional forms of quantification, largely from the field of stylometrics.

The last point in which our line of research diverges from current work in local coherence is the application of the metric, and as a result the method by which documents are compared. Whereas formal studies in local coherence are focused on determining whether an output contains an acceptable level of coherence, we are more interested in applying the knowledge gained from the metric to guide the text generation process itself. This tiered and interdisciplinary use of local coherence is an area that remains largely unexplored, but which can significantly contribute to the field of computational linguistics.

2.3 Rhetorical Figures as Discourse Entities

In Section 2.1 we focused our attention on the different techniques which may be used to identify the unique attributes of a document that can be used to formulate a metric for subjective evaluation of natural language processing systems. In this section we will shift our focus towards the application of these unique attributes. We identify what we will be measuring, how we intend to measure it, and how these measurements can be interpreted as a metric.

2.3.1 Theory of Rhetorical Figures

In order to find entities of discourse that match the criteria set in Section 2.2.3 we shall turn to the study of rhetorical figures. In this section we will examine the properties of rhetorical figures, and explain the qualities that lead us to selecting them as markers for our metric.

Classification

There are numerous ways of grouping rhetorical figures. For the purposes of this thesis we will consider three sorting categories: By method of deviation, by appeal (the area in which they apply their persuasion), and by their function.

Deviation: Schemes, Tropes, and Colour

The idea of figures of speech as deviations from the ‘norm’ is not a new one. As early as Quintilian [120] [119] figures of speech have been considered to be diverging, be it accidental or on purpose, from the ‘natural or proper’ patterns of speech [50]: They are thought of as “a rational change in meaning or language from the ordinary and simple form” [29], as “a forme of words, oration, or sentence, made new by art, differing from the vulgar maner and custome of writing or speaking” [111]. This way of thinking however is limited, as figures are an intrinsic element of language. From Aristotle in ancient Greece to the 18th Century [50] that same rejection of the notion that rhetorical figures are deviations has been repeated in some form or another: “There is nothing so natural, so ordinary, and so common as figures in human language.” Fontanier [60] proposes an alternate definition of figures. The deviation from a ‘simple’ form can be thought of as the form in which a phrase can be substituted with a more ‘straightforward’ one, even though the deviate may be statistically more common. In particular, we can draw an analogy between ‘deviates from simple’ and ‘has low semantic relatedness’.

Despite deviation being an overloaded term in the sense that we find it difficult to define the norm, we can still use it for categorisation, not by the degree of deviation, but by the form in which the figures manifest themselves.

Figures of Context: Colours

Colours are the least commonly discussed type of figure. Their characteristics, like their discussion, are nebulous at best. The term itself can be misleading. Colours have nothing to do with the chromatic attributes of a text; Instead, their occurrence is centered around their effect on the context of phrases, or “(the relations) between a sign and the elements which surround it within a concrete signifying instance”

[129]. In computational terms, this attribute of colours makes them very hard to automatically detect in text, since there are no rules or syntactic markers that can be exploited for the purpose of identification.

Figures of Syntax: Schemes

***S**ally **s**ells **s**eashells by the **s**ea **s**hore.*

Figure 2.1: Alliteration, a scheme which involves the repetition of the same letter or sound within nearby words.

Schemes are a particularly easy form of figure to detect. In the context of semiotic analysis [32] [33], schemes work along the syntagmatic axis of the semiotic plane [47] [46]. They are naturally rooted in the syntax of natural text, and are as a result qualities that arise from the arrangement of tokens and the selection of lexical alternatives. In other words, they are patterns, and computers are very good at handling that kind of information. In particular, schemes partially overlap with the field of natural language generation: Aggregation, coreference resolution, and referring expression generation, among others, all deal with the need to alter or improve the syntactic structure of text without changing the meaning [43] [66] [71] [98] [99] [3].

Figures of Meaning: Tropes

*He was excited **like** a weasel in a candy shop.*

Figure 2.2: Simile, a trope involving explicit comparison.

Whereas colours are of an ethereal nature and schemes are deeply rooted in syntax, the last category of divergence, tropes, takes the middle ground. Tropes deal with a change in meaning, and are to be considered as a semantic function of natural language text. In semiotic theory they take the paradigmatic (associative) axis of text analysis. The main function of tropes is to substitute, to separate the literal an interpreted meaning of figures. Most theories of rhetoric that consider figures of speech as deviation from the 'norm' have tropes in mind. Indeed, the most common types of tropes, the four 'master' categories [25] [26], are the ones most commonly associated with the notion of figure of speech: metaphor, metonymy, synecdoche, and irony.

Tropes are agents of paradigmatic change. In order to be able to fully exploit their variety we require a method of analysing their semantic content. In Section 2.3.2 we will examine how to take advantage of existent research in semantic analysis to effectively detect and classify tropes.

Appeal: Logos, Pathos, Ethos

Appeal is, in and on itself, not a true classification of figures. However, we deem appeal to be just as important to the success of this research. In rhetoric, persuasion is divided into three categories, depending on the target of the oratory: Logos, the appeal to reason, pathos, the appeal to emotion, and ethos, the appeal to character [124] [85].

In persuasion, logos is the is the category on which the argument is built. In fact, argumentation theory is almost exclusively focused on the appeal to logos [140] [141]

[8].

Will someone think of the ***children****?!*

Figure 2.3: An appeal to pathos

Cicero promoted the idea of pathos being used at the conclusion of the argument [36]. While Aristotle preferred that arguments be based as much as possible on logos, he did pay considerable attention to the appeal to emotion, and discussed it extensively in his *Rhetoric* [124] [85]. Despite being largely criticised, the study of pathos in rhetoric presents interesting opportunities for the application of the work in this document. Logos, on the other hand, is critical to public oration, particularly in judiciary settings. Still the rhetorical figures present in a text can alter the emotional state with which the reader confronts the information present in said text. By understanding how figures can be used to appeal to the audience's pathos we can guide computational linguistics tools the basis to construct documents with particular emotional appeals.

Ethos, the appeal to character, plays the most critical role during the first part of discourse. In order to make effective use of logos and pathos, the orator first needs to establish his/her credibility with the audience. We will not consider ethos for the scope of this study, as it is a field of significant debate, and there are no known methods of objectively determining the ethical quality of a text.

Function

Whereas deviation and appeal are very general means of classifying figures of speech, there are more fine-grained associations available. In fact, classification of figures by function is one of the most solidly established fields in rhetoric, having been covered throughout the ages by numerous authors [81] [119] [36] [111] [147] [114] [138]. In studying figures by function we can view the narrative as being ornamented and armed with the figures used throughout the narrative [55]. This view of rhetorical figures as tools is most relevant to this thesis. By considering rhetorical figures as salient features of a document we can use them as the entities for determining local coherence. Even more importantly, the classification of rhetorical figures by function arms us with guidelines for substitution, giving us a key tool to using local coherence as a guiding metric for natural language generation.

2.3.2 Semantic Analysis

A cardinal requirement for the work done in this thesis is that there be a method for automatically or semi-automatically identifying rhetorical figures in generated text. When discussing tropes we mentioned that to understand them we need to be able to understand the semantic aspects of rhetorical figures. It is for that reason that we turn to semantic analysis.

Lexical Semantics has been a growing field in recent years, boosted by continued development in lexical ontologies which allow researchers to exploit their structures and relations. The principal topics of research in the area of Lexical Semantics are semantic similarity and semantic relatedness. Work on these two topics is performed almost exclusively via the use of the WordNet lexical ontology [104], which has grown to be de-facto platform for research on word and word-sense relations. Other recent studies in similarity indices delve into the field of information retrieval, most notably exploiting popular search engines such as Google [19] [37] [12] and online collaborative encyclopaedias like Wikipedia [22][62] [125]. Some of this research is then applied to improve work in information retrieval, as shown by various researchers in the field [93] [73] [80].

Notable among non-semantic relations is the study of distributional similarity as proposed by Mohammad and Hirst [107], who eschew the use of WordNet and instead takes a corpus-centered approach and statistical measures. Some modern approaches by Fand and Friedman [57], Shimizu et. al. [128], and Cai and van Rijsbergen [30] use a combination of WordNet and corpus-based similarity measures.

In this thesis we will focus solely on similarity measures based on WordNet, since we are most interested in the explicit relations featured in it.

WordNet

WordNet [104] [102] [58] is the result of linguistic and psycholinguistic research at Princeton University. It is a lexical ontology for modelling the English language. In WordNet, information is structured based on word meaning, as opposed to more traditional ontologies that use the word forms for indexing. As a result, the function of words is reduced to that of a mere label, while the importance of meaning is elevated to the core position of classification. In WordNet, the concept of 'synset' embodies this classification, and all other relations are built on top of it. The central relation in synsets is synonymy, as a weakened form of the definition attributed in classical literature to Leibniz. By 'weakened' we mean that the definition of synonymy used in WordNet differs from the sense of 'true' synonymy (whereby two words are synonyms if they are always interchangeable in a sentence without altering the truth value of said sentence) by allowing near-synonymy, whereby substitution is not limited to single words. As a result, synonymy is defined as instances where the substitution of *expressions* in sentences does not alter their truth value.

Having thus defined the central building blocks of WordNet, we can turn our attention to how these synsets interact with each other. Literature on WordNet-based metrics identifies two major camps: semantic similarity and semantic relatedness [122] [24].

Semantic Similarity

Similarity in a lexico-semantic context is commonly denoted by the level in which concepts resemble each other. It can be thought of as feature commonality or feature overlap [136]. It is a strong quality in that common features increase similarity, whereas diverging features decrease similarity [91] [143]. Semantic similarity via the use of feature commonality and feature overlap shares some methods with stylometrics in that both shared as well as peripheral markers are used.

Edge-counting is the earliest and most common approach to measuring semantic similarity [117]. It focuses entirely on hypernymy/hyponymy (IS-A/SUBSUMES) relationships. A typical example of the edge-counting method is presented by Leacock and Chodorow [89]. The authors base their entire similarity measure on path lengths in the hypernymy/hyponymy network in WordNet. Similarity is then represented on a scale from zero to one as a function of the length of the path, with a longer path giving a lower score. The measurement of semantic similarity as a function of path length in WordNet has seen strong criticism. The major argument against it is that the accuracy of this metric depends on the density of the relational network. In particular, a setting with irregular densities (like WordNet) will produce inconsistent results in sparse elements of the network.

As an answer to the criticism of edge-counting, an alternative approach was proposed, pioneered by Resnik [122] and quickly adopted by the community. The proposed approach has come to be known as node-counting, and focuses on employing knowledge from external sources, such as corpus statistics, to provide a similarity index on top of the given taxonomies in WordNet. Lin [91] proposed a generalisation of the method to remove domain sensitivity from this approach. Central to their argument is the notion that similarity is bound not only by common features, but also by dissimilarities [143]. Most importantly, identical terms should be weighted equally, no matter how many commonalities they share. Critics of the node-counting approach argue that even though edge-counting is unreliable, it still provides a wealth of information that is ignored by node-counting methods. It has also been debated that stochastic, corpus-based methods require training, which is detrimental in that it requires a significant time investment and is subject to bias via corpus selection. As a result, hybrid methods have sprung up to fill the cleft between the two fields.

Hybrid counting methods, such as those of Jiang and Conrath [79], combine the strengths of both previous approaches. Edge weights are used as a primary metric, with corpus statistics applied as edge weights to correct inaccuracies that would emerge from uneven network densities. Yang and Powers [149] further improved these methods by increasing the network density through the inclusion of other relations present in WordNet beyond hypernymy/hyponymy. Finally, Pirró and Seco [113] expand on the earlier work of Seco, Veale, and Hayes [127] to propose an alternate corrective method to edge-counting by utilising the inherent structure of relation networks in WordNet. Under this method a greater number of hyponyms indicates that a term is less specific. The resulting metric is completely corpus-independent.

Semantic Relatedness

Semantic relatedness is a newer approach to relational metrics in WordNet. Unlike semantic similarity, which focuses on the hypernymy/hyponymy (IS-A) network of synsets, relatedness in a lexico-semantic context exploits as many relation networks as possible, including similarity. Semantic similarity can thus, in a way, be seen as a subset of semantic relatedness.

As with semantic similarity, work in the field of semantic relatedness started with edge-counting. Among the pioneers in this method was Sussna [136], who tapped the structures of semantic networks in WordNet for word sense disambiguation and information retrieval. However, unlike semantic similarity, where a single relation network is used, the edge-counting approach to semantic relatedness requires that the algorithm weigh paths on separate semantic networks differently. Among the two proposed weighing schemes is one not unlike that presented by Seco, Veale, and Hayes [127], namely that the strength of association for a particular term in a network is inversely proportional to the number of semantic relations for said term. In a similar fashion, the other weighing scheme argues that the closer a node is to being a leaf in the semantic network, the stronger its association with siblings is.

A different approach to path-centered semantic relatedness was proposed by Hirst and St-Onge [72]. Here, the authors postulate that the relations in WordNet can be classified by their type, 'upward', 'downward' and 'horizontal'. Relationship that can be classified as hierarchical (e.g. hypernymy/hyponymy) fall under the upward and downward categories, all others under the horizontal category. Using these classes of relations, the authors then formulate a set of predefined 'paths' for which relations are allowed. These paths are then used to determine if there is a significant relation if no previous 'strong' relation (synonymy, antonymy, or compounding) is present.

Path-based approaches to semantic relatedness have in the past suffered from lower performance against human controls when compared to semantic similarity metrics [24]. Scriver [126] has improved the performance of network-centered metrics of semantic relatedness by systematically decomposing them, eliminating the elements which turned out to be detrimental, and generating a simple formulation that proved to match the performance of Yang and Powers [149] but using an algorithm operating at lower complexity.

As an alternative to network-based metrics, Banerjee and Pedersen [7] proposed a radically different method of classifying relations. Instead of looking at the relation networks and hierarchies in WordNet, they focused on a previously-disregarded source: The glosses of synsets. The authors reckoned that there would be a link between gloss relatedness and synset relatedness. They not only counted the number of overlapping words, but also scored higher those glosses for which common terms shared similar ordering. This metric was further expanded by including synset neighbours in the gloss searches.

Applications and Limitations

We know that WordNet defines relations between synsets as networks of characteristics. Key among them are hypernymy (IS-A) and its counterpart hyponymy, which account to 65% of relations between noun synsets, holonymy (PART-OF) and its counterpart meronymy, and derivation. Furthermore, WordNet 3.0 introduces the INSTANCE-OF/HAS-INSTANCE relation. These relations can be exploited to give a measure of relatedness and parallelism between figures, which we can then translate into a metric of coherence in a fashion similar to the way referring expressions indicate salience in current entity-based studies of local coherence.

Many rhetorical figures can be mapped one-to-one or without much alteration to the relation networks present in WordNet. Take the example of synecdoche [119] [36] [111] [114], whereby "a whole is represented by naming of its parts (...), or vice-versa" [28]. This relation between the whole and its parts is clearly embodied in holonymy/meronymy. We can thus infer that it is possible to use semantic relations to determine the *type* of figure present. The similarity metrics presented in this section can be adapted as a confidence measure by which we can judge how much a section of text resembles a rhetorical figure. The main task yet to be resolved is the actual identification of figures.

2.3.3 Detection and Classification of Rhetorical Figures

Having identified the desired discourse entities and method for understanding them, we can now focus our attention to the actual task of detecting and classifying the rhetorical figures.

Detection

The method of detection of figures of speech will vary depending on the type of figure used. Most of the work on rhetorical figures in computational linguistics is focused on generating and/or understanding them, steps which are not necessary for the work presented in this thesis. Some work on detecting rhetorical figures has been described in Gawryjolek's work on annotation and visualisation of rhetorical figures [64]. However, we propose a method of rhetorical figure detection that differs from the ones presented in by Gawryjolek. We choose to focus on the use of semantic analysis to detect tropes. The major new contribution in this area is that the presented method is capable of detecting rhetorical figures without the need to understand them. Our focus on the use of semantic analysis makes figure detection possible because the saliency of rhetorical figures does not depend on the meaning of their constituent lexemes, but on their classification.

In order identify tropes, we first narrow the selection by doing a keyword search on tokens common to the figures, if available. For example, the 'is a' phrase is common to metaphors, as in 'Life is a highway' [39]. Next, we search for semantic markers that would indicate the presence of a figure. Continuing with the previous example, the semantic relatedness between 'life' and 'highway' is significantly lower than, say, 'A highway is a road'. Most schemes can be identified via pattern matching, be it structural (e.g. isocolon), token repetition (e.g. anaphora), or a combination thereof (e.g. epistrophe). It may be recommended to select preferred pattern matching engines for each kind of figure. Regular expressions, hidden Markov models [54] [115] [116], rule-based parsers [15], corpus-based machine learning [16], lexical chaining, or transformation-based error-driven learning [17] [18] are all candidates. Some may better fit patterns that maximise semantic similarity (e.g. diazeugma) or semantic distance (e.g. antithesis), suggesting that a hybrid engine, which would begin selection with either manual annotation or with pattern matching and refine searches with semantic measures, may be the best approach.

Classification

By large, most of the work required for classification of rhetorical figures will be done during the detection phase. Once a particular type of figure has been detected, it can easily be sorted by function and deviation type. One area where additional classification will be needed is when the figurative type of a discourse entity is ambiguous, such as figures that share common keywords. In those cases it will be necessary to examine the context of the entity (that is, its immediate neighbourhood) to disambiguate its type. In most cases a regressive examination of entities should suffice, under the assumption that preceding figures of speech have a stronger correlation with the examined entity. Where regression is not possible (such as no previous entities existing) the disambiguation can be delayed until the succeeding entities are annotated.

2.4 Classification

2.4.1 Document Classification

A significant open problem in the area of Computational Linguistics is that of automatic document classification. The goal is straightforward: Given a document and several categories, automatic document classification attempts to determine which category best fits the document. The history of automatic document classification is nearly as long as that of natural language processing, with Borko and Bernick [13] being among the pioneers in the field. As the data storage capacity of computing systems increased, so did the the number of electronically encoded documents. Classification is strongly relevant to information retrieval, as document classification is useful in indexing and searching for additional relevant documents.

Common methods of document classification use a stochastic approach, such as Bayesian networks [97]. Other approaches are nearest neighbour classification, decision trees and subspace method [90]. We propose that the rhetorical figures present in natural language documents can be used as a method of document classification which is both independent and complementary to existing algorithms.

2.4.2 Intent Classification

In Information Retrieval studies, the term ‘intent classification’ is generally used to label the classification for search queries. The field of Information Retrieval has bloomed with the advent of the Internet, and the rapid growth of readily-available but poorly-indexed documents. Query intent classification has many things in common with document classification, such as determining topic relevance [83] and nature of the query (e.g. ‘informational’ vs. ‘navigational’)[21][78].

However, there is little work done on classifying the intent of *user-generated content* as a metric. Instead research is generally focused on user-agent interaction, whereby the agent tries to infer the intent of a user, usually through a set of queries or dialogues [84] [92]. For content classification it is assumed in many cases that the publishing medium is enough to determine intent. However, the proliferation of user-content management and aggregation sites such as online journals, blogs, and social networking sites generate a melting pot of content that offers rich opportunities for classification strategies. Blogs in particular generate the most attention in this regard, and some research has been initiated in terms of sentiment classification [34][105].

2.4.3 Summary of Research Problems

In this chapter we have found unattended problems of metric-generation in natural language processing. We have focused our attention around those areas and identified the fields of study which we intend to combine in order to solve those problems.

We propose that specific rhetorical figures can be tied to particular forms of context, and that it is possible to examine the rhetorical figures of documents to generate enough context information to formulate a classification strategy for document intent. We suggest that this strategy is useful in Information Retrieval fields such as document classification, intent classification, and search optimisation.

Chapter 3

Domain Analysis

In this chapter we will explore the domain in which our study on detection and classification of rhetorical figures is set. We focus this Chapter exclusively on detection and preparation of rhetorical figures. Chapter 5 will have classification of rhetorical figures as its focus.

The detection and preparation of rhetorical figures requires a set of preliminary tasks. These tasks are the definition of the input corpora, the selection of a rhetorical figure to detect, and the identification of the attributes which shall be recorded for said figure. The last task can be divided into further sub-tasks - the identification of implicit versus explicit attributes, and the identification of intrinsic versus extraneous attributes. Explicit attributes are those stated in the formal definition of a rhetorical figure, whereas implicit attributes are not formally stated but emergent from the definition of the rhetorical figure. Similarly, intrinsic attributes define the rhetorical figure itself, and extraneous attributes are part of the figure's context. In this Chapter we will first examine the explicit intrinsic attributes of our rhetorical figure of choice. In Chapter 4 we will further examine the figure's implicit and extraneous attributes.

3.1 Preliminaries

A broad domain for epanaphora detection is undesirable as it introduces too many variables. The most effective method of constraining the research domain is to constrain the detection and classification of rhetorical figures to a single figure. This allows us to not only limit the number of variables involved in detection and classification, but it also enables us to proceed with a depth-first approach on rhetorical figure research instead of a shallow and superficial examination of numerous figures.

Having decided that the most efficient approach to constraining the domain for rhetorical figure detection is to restrict it to a single figure we needed to choose said rhetorical figure. In this thesis we chose rhetorical anaphora as our champion figure. Rhetorical anaphora has no direct relation to linguistic anaphora. Linguistic anaphora is commonly referred to simply as 'anaphora' in the domain of natural language processing. In order unambiguously to differentiate between the two concepts we will refer to rhetorical anaphora by one of its other common labels, epanaphora.

As mentioned above, restricting our domain to epanaphora enables us to perform a focused, in-depth study of that one rhetorical figure. Concentrating our initial study on just one rhetorical figure will give us the ability to refine our methodologies and improve the performance of our system on a well-studied task with a predictable rhetorical figure before expanding into more complex research.

3.1.1 Epanaphora

Definition

The commonly used definition of epanaphora, or rhetorical anaphora, is very simple:

> ***Epanaphora:*** *Repetition of the same word or group of words at the beginning of successive clauses, sentences, or lines.*

Identifying *any* instance of epanaphora in documents is a fairly straightforward task, as long as a parser and/or tokenizer are available to split the document into paragraphs, sentences, and words. However, the tasks of parsing and tokenization, while being able to reach near-human or in some cases better-than human accuracy, are not 100 percent accurate. This means that some errors may be introduced by these tools. We will address this issue later in this chapter. More important for the detection and classification of repetitions as epanaphora is that not every repetition is intentional, and not every intentional repetition has the primary goal to act as epanaphora.

Identification

Determining the intent of a rhetorical figure is not a well-defined problem. However epanaphora is classified as a *figure of repetition* by rhetoricians. We shall therefore equate the intent of epanaphora with the intent of repetition as a rhetorical strategy. Burton describes repetition as follows:

> ***Repetition:*** *"[A] major rhetorical strategy for producing emphasis, clarity, amplification, or emotional effect."* [28]

We can therefore label the intent of epanaphora as a strategy of repetition. Of particular interest are the keywords used in the definition, namely *emphasis*, *clarity*, *amplification*, and *emotional effect*. The use of these keywords provides strong hints towards the purpose of epanaphora as a rhetorical strategy. Having thus narrowed down the intent we can begin the work on generating a method for detecting, classifying, and sorting instances of epanaphora in real text as a function of a strategy of repetition. We will use three primary classifications for detected epanaphora in text. Each classification observes a different aspect of syntactic repetition versus repetition as a rhetorical strategy. These aspects are as follows:

- Intentional with primary anaphoric goal,
- intentional without primary anaphoric goal, and
- unintentional.

Unintentional epanaphora are accidental repetition, or repetition with very low confidence. The latter can occur when repetition is present in a paragraph, but the coherence between sentences is below a certain threshold. For example, sentences at opposite ends of a large paragraph share low repetition coherence because the elements of the repetition are spaced far apart. This category is primarily reserved for instances where an automated parser may detect a repetition but a human evaluator would not.

Intentional epanaphora without primary anaphoric goal denotes the instances of epanaphora where repetition is present and of high confidence, but where the primary goal is not emphasis by repetition. One example of such intentional epanaphora

You can either increase the power, or you can increase the speed.
If you increase the power, then you risk damaging the substrate.
If you increase the speed, then you risk overheating.

Figure 3.1: Anaphora with a repetition pattern.

Hurrah for love!
Hurrah for hope!
Hurrah for industry!
Hurrah for bonnie Canada, And her-bonnie maple tree!

Figure 3.2: Intentional rhetorical anaphora.

without primary anaphoric goal is antithesis (Figure 3.1). While antithesis does not require repetition, the latter is still common among instances of antithesis, as figures of contrast are in many cases built upon a framework of repetition. This type of repetition is conspicuous to a human evaluator, but does not fit within the constraints set by the definition of repetition as a rhetorical strategy.

Lastly, *intentional epanaphora with a primary anaphoric goal* are those repetitions that are conspicuous to human readers and whose intent falls within that of repetition as a rhetorical strategy. Figure 3.2 shows an example of rhetorical anaphora classified as intentional.

3.1.2 Data-sets

We used two input sets for epanaphora detection and classification. The criteria for using two different data-sets in our research was to prevent training bias from setting in. We divided our research into three distinct stages: discovery, advancement, and finalisation. These three stages roughly correspond to Chapters 3, 4, and 5 respectively. In terms of differentiation by procedures the discovery stage is primarily driven by human evaluators, the advancement stage is driven by human-supervised computational methods, and the finalisation stage is driven by chiefly automatic computational methods.

For the initial discovery stages we used exclusively the Blog Track from the 2006 Text REtrieval Conference (TREC)[109]. TREC was created to provide the infrastructure necessary for comparing information retrieval and document ranking methods on a large scale in a way that is easy to reproduce and evaluate. It consists of 100,649 unique entries crawled from RSS and Atom feeds, producing over three million documents. We focused on the TREC Blog Track as it provides us with training and testing sets which are no domain-specific and are representative of commonly-used prose of contemporary times. We deem both of these qualities to be of key importance in the training set. By using domain-independent training data we can ensure that we produce a domain-agnostic evaluation tool. By using text that is representative of modern prose we are able to determine whether the prevalence of certain rhetorical figures passes the test of time. To do so we can compare the results from running our methods against modern prose, and the outcome of performing evaluation against historical documents such as Victorian gazettes and Shakespearean plays.

In the advancement stage we begin with the aforementioned TREC Blog Track corpus. Towards the end of the stage we transition over to a corpus of Canadian 19^{th} Century literature [38]. This corpus contains over 56 thousand unique pieces of literature, digitised and converted to text using optical character recognition. During the rest of this thesis we will refer to this corpus simply as the *literature* corpus. This source is significantly different in style than the TREC Blog Track corpus. Since the *literature* corpus is based on print media it is more heavily-edited than self-published blogs. This is most evident in the extensive and structured style with which the content of the *literature* corpus is presented. In contrast the TREC Blog Track source features more compact segments of text with loose coupling between each segment.

The transition to the *literature* corpus in the advancement helped point out which aspects of epanaphora detection and classification are subject to bias arising from performing both training and evaluation on the same corpus or on split corpora from a single source. In order to maintain this separation of training and evaluation corpora, the finalisation stage was focused primarily on the *literature* corpus. The use of the TREC Blog Track corpus was limited to regression testing of any variation of the classification methods.

3.1.3 Text Handling

In Section 3.1.1 we explained that detection of epanaphora in text depends on properly identifying the boundaries between paragraphs, sentences, and words, but that existing methods are not one hundred percent accurate. Our algorithms for detection and classification are not solely dependent on achieving absolute accuracy in boundary detection, as long as the tokenization process is consistent. However, we do wish to limit the error rate in boundary detection to reasonable margins.

Of the three boundaries that need to be identified for epanaphora detection and classification, the easiest for us to identify are word boundaries. English convention dictates that words be separated by spaces and/or punctuation. Word boundary detection is the one area where our detection and classification algorithms are most resilient, since all but one of our criteria are content-agnostic when it comes to words. This demonstrates the advantages of performing a preliminary analysis on the domain of research before choosing the detection and classification criteria. By using specific sets of words as the basic tokens for sentence comparison we are limiting the requirements for accurate comparison to consistent and *sufficiently granular and atomic* tokenization. 'Sufficiently granular and atomic' refers to finding tokens whose size is appropriate to the task in which they are used. Where the task is epanaphora detection and classification single characters are too small, and entire sentences are too large. The best middle-ground is the tokenization of sentences into words based on separation by spaces and punctuation At the very least this token size is sufficient for our purposes. However we argue that using words as the base tokens is also ideal, since their boundary detection is simple, thus minimising the boundary detection error rate.

Paragraph Detection

To detect the boundaries between paragraphs in our input text we defer to the rich markup of the original document. We take the given paragraph boundaries as defined by the author (in the TREC Blog Track corpus) or by the optical character recognition software employed (*literature* corpus). The use of externally-defined boundaries gives us the flexibility of dynamically determining the scope of paragraphs based on

Worcester, Mass. Cottage ($I4.02)...........Portsmouth, N. H.
Danbury...............Danbury, Conn. Eastern Maine General...

Figure 3.3: Detection of abbreviations, decimal numbers, and ellipsis-like sentence non-terminators.

cues from the input text. It would thus be trivial to switch the paragraph detection to different markers based on the needs of our algorithm. In fact the *literature* corpus further limits paragraphs to text within a single page. The reason for this limitation is the inability of the character recognition software to differentiate between the headers, footers, and main content of each page. Since many of the attributes for epanaphora detection and classification depend on text continuity we decided to limit the scope of our input corpora in favour of more internally-coherent paragraphs. It is another statement in favour of the resiliency of our algorithms that this approach has shown no signs of significant loss in accuracy.

Sentence Boundary Detection

The last boundary detection problem for epanaphora detection and classification, sentence boundary detection is a hard, non-trivial problem in natural language processing. It plays a primary role in many aspects of NLP because for much of the work in natural language understanding the first step is sentence boundary detection. Some areas such as automatic summarisation, language models, and sentence alignment rely heavily on accurate sentence boundary detection.

For the initial purpose of this study we used a naïve blacklist approach to sentence boundary detection by splitting text at common punctuation characters: ".",

"!", and "?". We then created a set of whitelist rules and dictionaries to curb the incidence of false positive results in sentence boundary detection. In other words, the rules undo the sentence-splitting when it is deemed that the delimiters were not used as sentence boundaries. The most common occasions for such incorrect sentence boundaries are abbreviations ("Mr.", "ave.", &c.), ellipsis (. . .), and decimal numbers. Ellipsis and ellipsis-like forms, single-character abbreviations, and numeric values were targeted by generalized preset rules in the form of regular expressions. Ellipsis detection looks for adjacent sentence terminators. Non-printable characters are not counted against adjacency. Decimal numbers, on the other hand, are required to be adjacent to the decimal period on both sides of said period. Single-character abbreviations are also expected to be immediately adjacent to periods, but only on the left of the period. An example of text with ellipsis-like non-sentence terminators, a decimal number, and single-character abbreviations is shown in Figure 3.3. More abbreviations were handled by custom dictionary entries that were built as the abbreviations were encountered. The abbreviation detection process was sped up by dedicating a portion of our efforts towards explicitly training the dictionary of abbreviations. We performed sentence detection on the corpus, reviewed the sentence boundaries, and added any abbreviations that were found during the review to the dictionary. The process was then repeated multiple times until the incidence of abbreviation boundaries fell below a threshold deemed acceptable.

We argue that a naïve implementation of sentence boundary detection like the one described above is sufficient to accurately generate results via our algorithm. Previous implementations of regular expression-based systems show that with some tuning a simple system can achieve 99.1% accuracy [1]. More complex systems offer comparable accuracy, with error ratios of 0.52% [65], 1.65% [86], 1.45% [101], 1.0% [110], and 1.2% [123]. The implementation of these systems varies (in no particular order) from SVMs to decision trees, part-of-speech lexicons and neural nets, maximum entropy, and detection of abbreviations via heuristics. Even without tuning, the measured baseline accuracy of 74% for naive sentence boundary detection without whitelists would have been acceptable for training our system. The reasons for the lower acceptable sentence detection accuracy are two-fold: First, most of the results given for the systems previously described were based on training and testing on domain-specific, well-formed text, namely the Wall-Street Journal corpus from Penn Treebank [106]. That corpus is not an accurate representation of general-domain text, and the error rates for tailored lists are significantly higher on TREC Blog corpora. Second, we can extrapolate some of our reasoning for using simple word tokenization to sentence boundary detection. For training, we were not interested in locating *every* occurrence of repetition, but in ensuring that the ones we do detect were appropriate. We therefore had the option to ignore the loss of results due to truncation on periods which do not occur on sentence boundaries. Nevertheless our whitelist-trained sentence boundary detector is capable of a measured accuracy higher than 95%, far above the initial baseline accuracy of 74% of an untrained system. This accuracy was measured by having our expert annotators tag sentences with incorrect boundaries during the early stages of training of epanaphora detection algorithms.

Finally, since paragraph detection, sentence detection, and figure detection all function independently from each other, our system is capable of individually adjusting the internal parameters for each of the tasks without affecting the execution of the others. It would thus be trivial to adjust the system to be able to handle domain-specific parses, such as treating lines of text as sentences in order to better detect epanaphora in poetry.

3.1.4 Attribute Exploration

The process in which we proceeded with the exploration of attributes of repetition began with recording a rough set of metrics on attributes of epanaphora-like repetition, followed by a consultation of expert authorities in the fields of rhetoric, rhetorical figuration, and epanaphora [70], [56], [144]. During the consultations a small set of predictions about the nature of intentional and accidental epanaphora were drawn up. The next step involved refining the metrics of attributes of repetition in order to verify or disprove these predictions. The evaluation of this last step was performed by hiring expert reviewers to perform a qualified refinement of the recorded attributes.

The remaining sections in this chapter are dedicated to the exploration of the intrinsic and explicit attributes of repetition. These attributes are the size of the repeating pattern and the length of the repetition. We will study the following aspects:

- The characteristics of each attribute,
- How the attributes are distributed among the corpora of intentional and accidental epanaphora,

- How this distribution correlates to the characteristics of the attributes, and
- How to ensure that all variants of an attribute's characteristics are recorded.

The first three aspects – characteristics, distribution, and correlation – are intended to foment the inclusion of human insight into the development of strategies for epanaphora detection and classification. The last aspect is necessary to ensure that the results from the previous aspects are comprehensive.

3.2 n-Grams

The most basic variation in detecting epanaphora is to observe the size of the repeating pattern. Increasing the size of the pattern is equivalent to increasing the number of tokens that need to be matched between sentences before considering them as matching our rules. Due to the nature in which the repeating patterns are recorded we will also refer to this technique as the recording of *n-gram overlap*.

3.2.1 Decisions Supporting the Use of the 'n-Gram' Label

The term *n-gram* refers to a subsequence of *n* items within a given sequence. More specifically, when labelling a subsequence as an n-gram we expect the items to be contiguous. It is important not to confuse our use of the term *n-gram* with models based on n-gram sequences, such as *n-gram models* which are largely used in statistical natural language processing [53][95] for tasks such as part-of-speech tagging via the use of hidden Markov models [44][54][115][116].

$$S_x = x_1, x_2, \ldots, y_n$$
$$S_y = y_1, y_2, \ldots, y_n$$

Figure 3.4: Overview decomposition of sentences S_x, S_y into their x, y tokens.

Our choice for giving the label of n-gram to the subsequences that we search for is founded on our wish to distinguish token subsequences (which occur across pairs of sentences) and collections of paired sentences within paragraphs. Since the former are gapless and sequential, we concluded that the best label choice is *n-gram*. The decision is further enforced by our need to classify the subsequences by length, since n-grams sport a convenient parallelism between the naming scheme and their length.

The symbol that we use to label n-gram overlaps in formulas will be η. For more information on the naming scheme for sentence groups, refer to Section 3.3.

3.2.2 Formal Definition of n-Grams as Used in This Thesis

We already mentioned that n-grams are subsequences of tokens within sentences, and that they are gapless and ordered. Given the definition for epanaphora, we can further narrow down the definition of n-grams in epanaphora by requiring that only leading paired tokens in each sentence be considered as an n-gram. In other words, only n-grams that form at the beginning of sentences are considered. The classification and labelling of the resulting n-gram will depend on the length of the matched token subsequence.

Identification of n-gram overlaps in pairs of sentences follows an iterative approach. Consider two sentences, S_x and S_y (Figure 3.4). When looking for matching

$S_x =$ *The deadline for* registration is Monday.
$S_y =$ *The deadline for* paying the registration fee is the Monday after.

Figure 3.5: Example of two sentences with a *3-gram overlap*.

n-grams, we begin at the first token of each sentence (x_1 and y_1, respectively) and proceed to compare the tokens pairs sequentially, comparing tokens with the same subscript. The pair comparison continues until a pair is reached where the x and y tokens are not equal (the first mismatch). The size of the *n-gram overlap* is determined by the number of matched pairs. For example, a n-gram comparison between S_x and S_y in Figure 3.5 shows that $[x_1, x_2, x_3]$ equals to $[y_1, y_2, y_3]$ but that $[x_4]$ is not equal to $[y_4]$, thus making the n-gram overlap between S_x and S_y is three. The steps required to locate and classify an n-gram overlap are crystallised in Algorithm 3.1.

The formal logic behind Algorithm 3.1 is simple. In general terms, the n-gram overlap η is equal to the size of the longest set of pairs $R = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ such that the rules in Equation 3.1 apply:

$$\forall (x_i, y_i) \bullet i = \{1, \ldots, n\} \land x_i = y_i \land x_{n+1} \neq y_{n+1} \tag{3.1}$$

The first two parts of the right side of Equation 3.1 ensure that all pairs in R are composed of equal values. The last part, $x_{n+1} \neq y_{n+1}$ indicates that the $n + 1$th pair is *not* composed of equal values. There are n (x_i, y_i) pairs in R, and we know that the involved tokens are consecutive. We can therefore state that

Algorithm 3.1 Find the longest n-gram overlap between sentences S_1, S_2

```
Require: $S_1, S_2$
$l_1$ := length of $S_1$
$l_2$ := length of $S_2$
if $l_1 = 0$ or $l_2 = 0$ then
    return 0
end if
$max := 0$
if $l_2 > l_1$ then
    $max := l_2$
else
    $max := l_1$
end if
$count := 0$
while $count < max$ and $S_1[count] = S_2[count]$ do
    $count := count + 1$
end while
return $count$
```

$$\eta = |R| \tag{3.2}$$

Furthermore we need to restrict R to non-empty sets ($R \neq \emptyset$). Expression (3.2) can thus be expanded as follows:

$$\eta = \begin{cases} 0 & \text{if } R = \emptyset \\ n & \text{otherwise} \end{cases} \tag{3.3}$$

3.2.3 Lower n-Gram Bounds

By introducing a limit on the minimum length of an n-gram overlap, we can examine the effect that such a lower limit has on the number of instances of intentional epanaphora. To add a lower limit we alter the conditions in Equation 3.3 to introduce a floor value:

$$\eta_{lower} = \begin{cases} 0 & \text{if } R = \emptyset \\ 0 & \text{if } n < b_{lower} \\ n & \text{otherwise} \end{cases} \tag{3.4}$$

Here we use the subscript *lower* to denote that the n-gram overlap η has a set floor value. We now change Algorithm 3.1 to match these requirements:

Algorithm 3.2 Find η between sentences S_1, S_2 where $\eta \geq b_{lower}$

```
Require: $S_1, S_2, b_{lower}$
 1: $l_1$ := length of $S_1$
 2: $l_2$ := length of $S_2$
 3: if $l_1 = 0$ or $l_2 = 0$ then
 4:     return 0
 5: end if
 6: $max := 0$
 7: if $l_2 > l_1$ then
 8:     $max := l_2$
 9: else
10:     $max := l_1$
11: end if
12: $count := 0$
13: while $count < max$ and $S_1[count] = S_2[count]$ do
14:     $count := count + 1$
15: end while
16: if $count < b_{lower}$ then
17:     $count := 0$
18: end if
19: return $count$
```

By adding the condition on lines 16-18 of Algorithm 3.2 we have thus set a lower bound requirement for our n-gram overlap. We initially speculated that a larger n-gram overlap can be mapped to a higher confidence that an instance of epanaphora is intentional.

Hypothesis 3.2.1. *The size of a n-gram overlap is proportional to the confidence that said n-gram overlap corresponds to epanaphora.*

To back up Hypothesis 3.2.1 we argue that a larger n-gram overlap reduces the likelihood of a repetition being accidental. That process should be particularly evident in text that has gone through an editing process involving either a human reviewer or automated text repair. Accidental repetition is commonly eliminated through techniques such as aggregation [43][118] and the introduction of referring expressions [2][3][42].

Correlation Between Lower n-Gram Bounds and Intention

The length of n-grams in sentence pairs can have a significant effect on quality of epanaphora identification. However, the length value alone cannot be relied on exclusively as a deciding factor. As Figure 3.6 shows, the number of paired sentences drops off very rapidly. This drop-off would be acceptable if the results for 1-gram- and 2-gram overlaps were almost exclusively unintentional. However, that is not the case, and even if one out of 100 results were intentional that number would be larger than the total number of results for 5-gram overlaps. Furthermore, we discovered that as we increased the bound b_{lower}, the perceived percentage of intentional epanaphora actually *dropped*. We conclude that this lowered percentage of perceived intentional epanaphora is the result of our choice of text source, particularly due to sentence duplication and to the use of the medium for spamming.

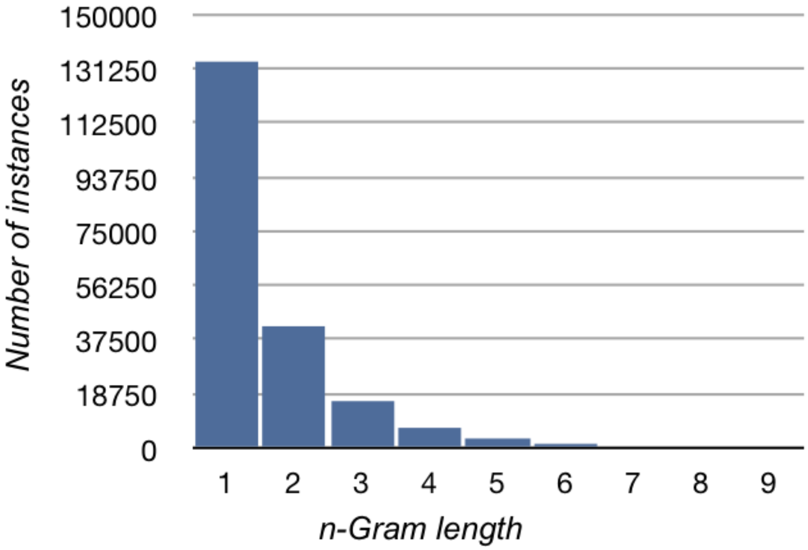


Figure 3.6: Instances of epanaphora at increasing lower bounds b_{lower}.

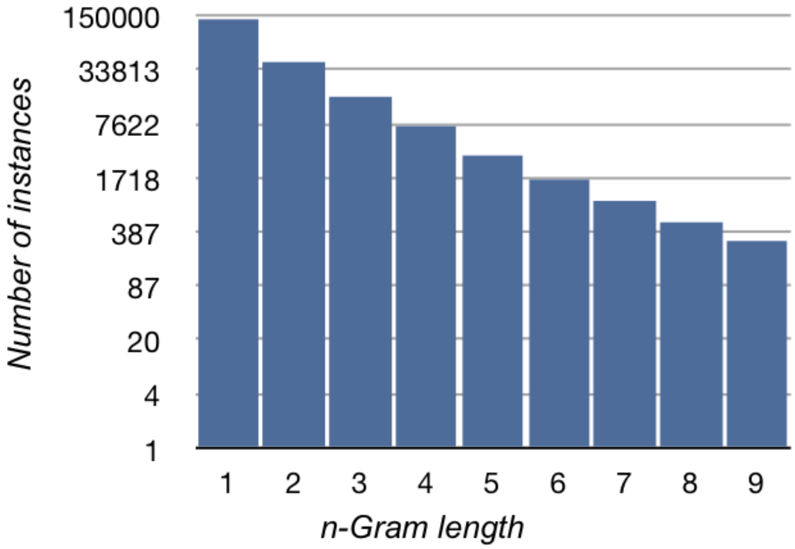


Figure 3.7: Instances of epanaphora where $\eta \geq b_{lower}$ (logarithmic).

We didn't get to the hotel in time to see the game [. . .].
We didn't get to use our reservations at the seafood buffet [. . .].

Figure 3.8: 4-gram overlap of semantically divergent sentences.

One type of accidental epanaphora that we had hoped to reduce is the use of divergent compound phrases. Figure 3.8 represents such a divergence. Each sentence uses a different sense of the phrase *get to* – the first sentence uses it in the context of movement, whereas the second sentence the phrase alters the tense of the verb *use*. We had intended to push these types of repetition off the results chart by using the lower bound, but the compound phrases occur too late in the sentence to be effectively removed.

Earlier in Section 3.2.3 we had argued that higher values for the bound b_{lower} indicate a better chance that a particular repetition is intentional. What we had not considered, however, is the effect of more literal repetition on the rhetorical value of the sentence pair. At constant average sentence length, a larger n-gram overlap reduces the variation space for the remainder of the sentences. Given a cluster of sentences of length n and an n-gram of length i, the variation space is of complexity $O(2^j)$ on the remaining j tokens in each sentence, where $j = n - i$. As a result, it becomes less likely for authors to produce significant (non-verbatim) patterns of repetition with larger n-gram overlaps. Our theory is that this effect is an important factor in explaining the rapid decline in instances of epanaphora as the size of the bound b_{lower} increases.

Finally, we need re-examine Figure 3.6. The values are deceptive because the number of found epanaphora at each step x include not only the instances of epanaphora with n-gram overlap x, but also all those of larger n-gram overlap. In effect, in Figure 3.6 the values for the bars at each step x follow formula 3.5:

$$f(x) = \sum_{i=x}^{n} \eta_i \tag{3.5}$$

In order to get a more accurate view of the size difference between the sets of epanaphora at each step we need to implement an upper bound b_{upper}.

3.2.4 Upper n-Gram Bounds

The focus on n-grams so far has been on putting a lower limit b_{lower} on the number of token pairs required to generate a desired n-gram overlap. However, in Section 3.2.3 we have laid out evidence that indicates that the majority of instances of intentional epanaphora are encountered towards the short end of n-gram overlaps. Figure 3.7 in particular tells us that the total number of instances of epanaphora decays exponentially. We theorise that this trend is mirrored by intentional instances of epanaphora. By introducing a ceiling boundary b_{upper} we can put in place an additional filter for the search range of η. We update Equation 3.4 and Algorithm 3.2 correspondingly.

$$\eta_{lower,upper} = \begin{cases} 0 & \text{if } R = \emptyset \\ 0 & \text{if } n < b_{lower} \vee n > b_{upper} \vee \\ n & \text{otherwise} \end{cases} \tag{3.6}$$

The important changes in Algorithm 3.3 are not only the introduction of the

Algorithm 3.3 Find a η for S_1, S_2 where $b_{lower} \leq \eta \leq b_{upper}$

```
Require: $S_1, S_2, b_{lower}, b_{upper}$
Ensure: $b_{lower} \leq b_{upper}$
$l_1$ := length of $S_1$
$l_2$ := length of $S_2$
if $l_1 = 0$ or $l_2 = 0$ then
    return 0
end if
$max := 0$
if $l_2 > l_1$ then
    $max := l_2$
else
    $max := l_1$
end if
$count := 0$
while $count < max$ and $S_1[count] = S_2[count]$ do
    $count := count + 1$
end while
if $count < b_{lower}$ or $count > b_{upper}$ then
    $count := 0$
end if
return $count$
```

upper bound b_{upper} and its inclusion on line 16, but also the requisite that b_{lower} should not exceed b_{upper}.

By setting an upper bound on the instances of epanaphora, we are preventing the results from η_i from being poisoned by those from η_j, where $i < j$. We have discussed in Section 3.2.3 how instances of epanaphora with large η are predominantly literal duplications. While the exponential decay in number of epanaphora instances means that these literal duplications have a significantly smaller effect as the gap between i and j increases ($\eta_i \ll \eta_j$), the methods described in this section let us mitigate the effect of those instances where i and j are close.

An additional advantage of using the bounds b_{lower} and b_{upper} is that we are not limited to searching for sentences of one particular length, but can instead search over a range of sizes for the n-gram overlap η. This variable search range can be used as an optimisation technique when there is no fixed target for the size of η, but some reasonable limits are known.

We can now address the concerns raised at the end of Section 3.2.3, namely the bias in Figure 3.6.

As Figure 3.9 shows, the number of instances of n-gram overlaps at each step also decays exponentially. The slight jump for $\eta = 9$ is due to the count at that value also including instances of larger size. What we did was set the upper limit such that $b_{upper} = \infty$, in effect acting as if there was no ceiling at all.

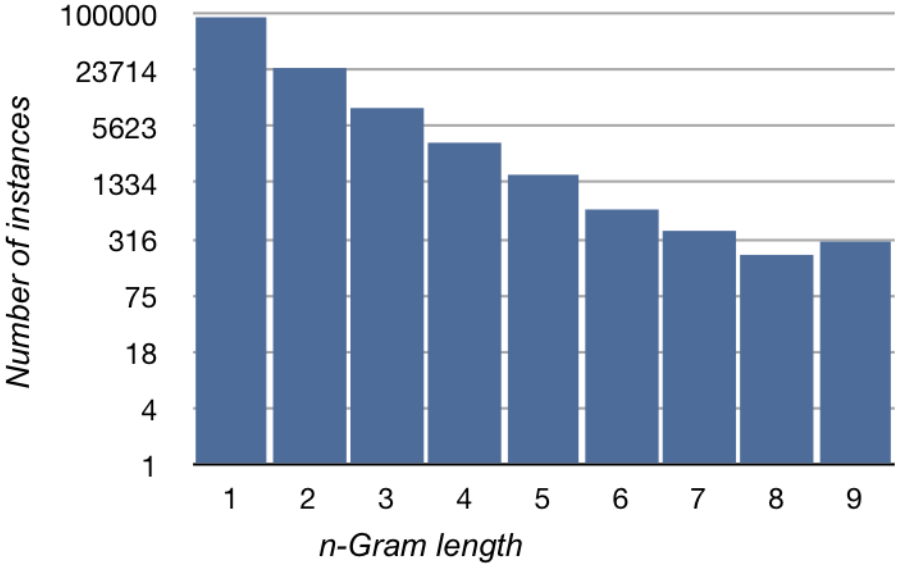


Figure 3.9: Instances of epanaphora where $b_{lower} = \eta = b_{upper}$ (logarithmic).

3.3 Tuples

The previous section has been focused on finding sequences of paired tokens between given two sentences. However, we have shown that n-gram overlaps are not sufficient to accurately identify instances of epanaphora. Many epanaphora span multiple sentences. In fact, we shall argue that a recurring pattern that spans multiple sentences has a stronger likelihood of being intentional than epanaphora that span only a single pair of sentences. In this section we will discuss our approach to sorting instances of epanaphora by their recurrence patterns. During the rest of this thesis we will refer to these recurrences as *sentence tuples* or just plain *tuples*.

3.3.1 Naming Scheme

In computer science and mathematics, the term *tuple* is assigned to ordered lists of elements [61][137]. The etymology of the word tuple comes from the abstraction of the words for collections of increasing sizes: double, tri*ple*, quadr*uple*, quin*tuple*, sex*tuple*, &c.

The key factor for using the term *n-gram* for subsequences of tokens in Section 3.2 and the term *tuple* for collections of sentences in this section is that, unlike the subsequences in n-gram overlaps, our collection allows gaps (unmatched sentences) to exist between the sentences comprising the collection. Since the definition of tuples in set theory does not require elements of a tuple to be contiguous (only ordered and typed), we judged that it is more appropriate for labelling the collections of sentences for epanaphora. There is one caveat: In set theory, tuples are allowed to contain more than one instance of an object. Normally, the term *set* is used for tuples without duplicate elements. However, we deem the term *set* to be too ambiguous in the context of this thesis, hence we shall not use that label. For the purpose of this thesis our sentence tuples shall be ordered collections of *unique* sentences. We define a sequence of tokens – sentences being sequences of words, and words being sequences of characters – within a document as *unique* if none of its constituent tokens are contained within any other sequence. That is, there is no such token x such that it is contained in both the sequence S_i and the remainder of sequences $S - S_i$.

$$\forall(S_i) \bullet i = \{1, \dots, n\} \wedge x \in S_i \wedge x \notin S - S_i \tag{3.7}$$

In contrast, two (sub)sequences S_1 and S_2 are *equal* if each element of the sequence S_1 has the same value as its counterpart in sequence S_2. In effect, the search for n-gram overlaps in Section 3.2 is a test for equality. By contrast, for two (sub)sequences to be unique, all their constituent elements must occupy non-overlapping ranges within the document. It is possible for two sequences to be unique but equal. If the sequences are sentences within a document, then this means that S_2 is a literal copy of S_1.

Figure 3.10 sums up and contrasts the attributes of *n-gram overlaps* and *tuples*.

Whenever we refer to a tuple in formulas, we shall use the symbol φ. In the next section we shall define how tuples are constructed and identified for each paragraph in a document.

n-Gram overlaps:	**Tuples:**
• Ordered	• Ordered
• Gapless	• Gaps allowed
• Tokens are not unique	• Unique tokens
• $x_1, x_2, x_3, x_4, \ldots$	• $y_1, \ldots, y_2, \ldots, y_3, \ldots$

Figure 3.10: Comparison between *n-gram overlaps* and *tuples*

3.3.2 Formal Definition of Tuples as Used

In our approach, searching for sentence tuples uses the same basic method regardless of the desired tuple size. The process has two primary steps: collection and identification.

Collection

Collection (recording) of tuples is performed pair-wise in an iterative fashion. The boundaries of the collection process are the boundaries of the current paragraph. We begin with the first sentence S_1 in the paragraph, and iterate over every sentence in $S_2, \ldots, S_n$ to find other sentences that can be paired with S_1. The next iteration begins at S_2, and proceeds over $S_3, \ldots, S_n$. In effect we are performing a comprehensive search for n-gram overlaps within each paragraph. This comprehensive search process is crystallised in Algorithm 3.4.

The set of pairs returned by Algorithm 3.4 is not yet a proper tuple. In fact, the collection of pairs returned on line 13 of Algorithm 3.4 has the potential of having $\lfloor \frac{n}{2} \rfloor$ tuples, if a paragraph consists of $\frac{n}{2}$ *unequal* n-gram overlaps. We will later show how it is actually possible to find $n-1$ distinct tuples in a paragraph if each pair

Algorithm 3.4 Find the comprehensive set of n-gram overlaps in paragraph P

```
Require: $P = S_1, \ldots, S_n$
if $n < 2$ then
    return [ ] {Empty collection}
end if
$pairs$ := [ ] {New ordered list}
for $i := 1$ to $n - 1$ do
    for $j := i + 1$ to $n$ do
        $r$ := FindNGram( $S_i, S_j$ )
        if $r > 0$ then
            $pairs \leftarrow [S_i, S_j, r]$ {Append $[S_i, S_j, r]$ to the $pairs$ collection}
        end if
    end for
end for
return $pairs$
```

in *pairs* has a different n-gram overlap size r. But first we need to understand the process of identifying a tuple out of the *pairs* collection from Algorithm 3.4.

Tuple Size Identification

Each entry in *pairs* consists of a sentence pair S_a, S_b and the size r of its n-gram overlap. We know that to form a pair from S_a and S_b, the size r of their n-gram overlap has to be non-zero. If the length of the collection *pairs* is one, then S_a, S_b is the only tuple in paragraph P. Since only two sentences are involved in this tuple, they are a *double*. Therefore, the (non-trivial) base case is that there is only one tuple in P, and that it is a *double*. Every step below this point will assume that the size of the set *pairs* is two or greater. Since we know that *pairs* has two or more entries we need to be able to distinguish between them. We shall label each of the entries in the set *pairs* as ρ, where $pairs = \rho_1, \ldots, \rho n$. Let us examine some of the properties of the elements of ρ.

We know that each ρ in *pairs* contains two sentences, S_a and S_b. Furthermore, we know that they have a certain order, as set in line 9 in Algorithm 3.4. This order is important to us. In particular, it guarantees that if the sentence order in element ρ_i is S_a, S_b, then there will be no element ρ_j where $j > i$ and $S_x, S_a \in \rho_j$. In other words, if S_a is the first sentence in ρ_i, then we are guaranteed to know that S_a will not occur as the second sentence in any later element ρ_j. This statement is simple to prove. Line 9 shows S_i as being the first element in any entry ρ. S_i depends on the outer loop on line 5. Therefore, any entry with S_i as the first element (S_a) of ρ **must** occur within this iteration of the outer loop. Furthermore, S_i will not be present in *any* entry ρ after this iteration of the outer loop. The absence of S_i in any ρ after the current iteration of the outer loop is because the outer loop will have advanced to $i + 1$, and because the inner loop on line 6 guarantees that j is greater than i. Thus, after the ith iteration of the outer loop, S_i will not occur again in any entry ρ in the *pairs* collection. □

Observation 3.3.1. *If S_x is the first sentence in ρ_i, then S_x is guaranteed to never be the second sentence in some ρ_j where $j > i$.*

Furthermore, we know that all entries ρ where S_i is the first element S_a must occur within iteration i of the outer loop in Algorithm 3.4, and that all entries ρ created during iteration i of the outer loop must have S_i as S_a. From that knowledge we can extract the following two observations:

Observation 3.3.2. *Any entries ρ_x, ρ_y where $S_{a,x} = S_{a,y}$ must come from the same iteration of the outer loop in Algorithm 3.4.*

Observation 3.3.3. *Take any two entries ρ_x, ρ_y where $x < y$ and $S_{a,x} = S_{a,y}$. For every other entry ρ_i where $x < i < y$ we can guarantee that $S_{a,x} = S_{a,i} = S_{a,y}$.*

We can also say something about the order of the second sentence, S_b, in each entry ρ that has the same first sentence, S_a. Since each S_a in these entries is the same we know from Observation 3.3.2 that the outer loop in Algorithm 3.4 must not advance among these entries. Therefore, the only change between these entries must come from the inner loop on line 6 of Algorithm 3.4. Since this loop advances j in a sequential order, we can guarantee that the location of $S_{b,j}$ in the entries ρ are ordered according to their occurrence within paragraph P. □

Observation 3.3.4. *For every entry ρ with the same first sentence S_a, the order in which the second sentences S_b appear in the paragraph determines the order in which the corresponding entries ρ appear in the collection.*

Finally, we can derive another observation from the collection $pairs$. Each entry ρ in the collection that has the same S_a must share at least one common tuple φ. It is a given that there is a non-zero n-gram overlap for each sentence pair S_a, S_b in these entries ρ. It follows that for each entry ρ, at least the first token in each sentence is equal. Furthermore, since S_a is present in every entry, the sentences in each entry rho must contain at least one equal token at the beginning of the sentence. This means that every of these entries ρ must share a tuple with the minimum n-gram overlap of one token. □

Observation 3.3.5. *For each set of entries ρ that share a common first sentence S_a, there must exist a tuple of size at least equal to the number of such entries ρ.*

We can expand on observation 3.3.5 and garner some additional knowledge about the maximum tuple size for any sentence S_x. We know that a sentence with a non-zero n-gram overlap against at least one other sentence must appear in any entry ρ of the collection zero or more times as S_a, zero or more times as S_b, and at least one time as either S_a or S_b. We also know that all entries where S_x appears as S_a are contiguous, and that there is no entry where S_x appears as S_b after any entry where it appears as S_a. Therefore we can conclude that any entry with S_x as S_b must appear in the collection before any entry with S_x as S_b.

Observation 3.3.6. *Any entry with S_x as S_b must appear in the collection before any entry with S_x as S_b.*

We know that all entries with S_x as S_a match S_x against sentence that occur after it in paragraph P. Since the inner loop prevents us from matching any sentence S_i against sentences that occur before it in paragraph P, we can thus deduce that any pairing of S_x against any sentence that occurs before it in P must be handled in entries where S_x occurs as S_b. If S_x occurs as S_b in any entry ρ, then it must be because some sentence S_y (where $y < x$) was matched against it during the outer-loop iteration for y. In addition, additional occurrences of S_x as S_b must be the result of a match against a different S_y. Each of these entries is unique. Therefore we conclude that each occurrence of S_x as either S_a or S_b is unique. Lastly, we can state that there is no sentence S_z such that S_x and S_z form a non-zero n-gram overlap, but where there is no entry ρ in the collection such that $S_x = S_a, S_z = S_b$ or $S_z = S_a, S_x = S_b$. In other words, a non-zero n-gram overlap between any two sentences S_x and S_z **must** generate an entry in the collection $pairs$.

Observation 3.3.7. *There is no sentence S_y such that the n-gram overlap between S_x and S_y is non-zero and there is no entry ρ in the collection where both S_x and S_y are present.*

Observation 3.3.8. *A non-zero n-gram overlap between any two sentences S_x and S_y must generate an entry ρ in the collection, such that $(S_x = S_a \wedge S_y = S_b) \vee (S_y = S_a \wedge S_x = S_b)$*

Therefore, any sentence that generates a non-zero n-gram overlap with S_x generates an entry where $S_x = S_a \vee S_x = S_b$. If we combine Observation 3.3.8 with Observation 3.3.5 we get the following:

Observation 3.3.9. *For each set of entries ρ that share a common sentence S_x where $S_x = S_a \vee S_x = S_b$, there must exist a tuple of size at least equal to the number of such entries ρ.*

Combining Observation 3.3.9 with Observation 3.3.7 we can also generate an upper bound for tuples involving S_x:

Observation 3.3.10. *For each set of entries ρ that share a common sentence S_x where $S_x = S_a \vee S_x = S_b$, the largest tuple possible tuple size involving S_x is equal to the number of such entries ρ.*

Lastly, we observe that, by selecting the appropriate S_x, we can optimise our search for maximum tuples. From Observation 3.3.6 we know that for $S_i = S_a \in \rho_1$ in the collection there are no entries for which S_i is S_b, and from Observations 3.3.7 and 3.3.8 we know that all non-zero n-gram overlaps in P generate an entry in the collection. Furthermore, we know that all entries for S_i are contiguous, due to Observation 3.3.3. Therefore, the subset of entries from the collection where $S_a = S_i$ is the largest tuple φ for the minimum n-gram overlap involving S_i. Any other entries that involve any sentence S_j of the S_b sentences from this subset must be paired with other S_b sentences, and cannot involve any sentence not matched against S_i. If there was any entry ρ_m such that $S_{a,m} = Sj$ or $S_{b,m} = Sj$ where S_j is one of the S_b sentences for S_i, but where $S_k = S_{b,m} \vee S_{a,m}$ is not one of the S_b sentences for S_i, then that would violate Observation 3.3.7 because S_k and S_i must share a non-zero n-gram overlap. We can therefore count all the entries for S_i, eliminate any additional entry involving each S_b for S_i, and repeat the process. Each time, the count represents the maximum size for a unique tuple in this paragraph.

Observation 3.3.11. *For $S_x = S_{a,1}$, the largest tuple involving S_x can be found by locating all entries in the collection that have S_x as their S_a.*

We thus have collected enough information to generate a list of unique tuples φ in paragraph P. as mentioned, the process to do so requires iteration over the collection of entries generated during Algorithm 3.4. The result is Algorithm 3.5.

Algorithm 3.5 breaks down as follows – lines 4 to 7 identify the first tuple φ_1 in collection C, based on Observation 3.3.11. This tuple is then stored for later perusal. In lines 12 to 20 we are locating entries in C that have no relation to the entries in tuple φ_1. We know that the first k entries belong to tuple φ_1; Line 13 bypasses these. By this point we also know that none of the remaining entries in collection C will have any sentence that is equal to S_a from any of the entries in φ_1, due to Observations 3.3.3 and 3.3.6. Therefore we only need to check the remainder

Algorithm 3.5 Find the unique tuples in collection C

```
Require: $C = \rho_1, \ldots, \rho_n$
if $n < 2$ then
    return $C$ {There is only one tuple, a double}
end if
i:=1
while $S_{a,i} = S_{a,1}$ do
    $i := i + 1$
end while
$tuples := [\ ]$
$tuple \leftarrow [\rho_1, \ldots, \rho_i]$
$tuples \leftarrow tuple$
$remain = [\ ]$
for $j := 1$ to $n$ do
    if $j > i$ then
        for $k := 1$ to $i$ do
            if $S_{a,j} \neq S_{b,k}$ and $S_{b,j} \neq S_{b,k}$ then
                $remain \leftarrow \rho_j$ {$\rho_j$ is part of a different tuple, store}
            end if
        end for
    end if
end for
$temp :=$ recurse( $remain$ )
for $m = 1$ to length of $temp$ do
    $tuples \leftarrow temp_m$
end for
return $tuples$
```

The deadline for registration is Monday. *The deadline for* paying the registration fee is the Monday after. **You** will not be able to register over the weekend. **You** may want to pay the registration fee at the same time as you register. **You** will be e-mailed your registration details upon payment receipt.

Figure 3.11: Two unique tuples in a paragraph.

of collection C for entries that contain any of the S_b sentences in tuple φ_1, which is done in Algorithm 3.5 on lines 14 and 15. Finally, line 16 is reached if and only if entry ρ_j is not related to tuple φ_1. We therefore store this entry in a separate list for recursion, being careful to preserve the order of the entries. The return value, *tuples*, is the collection of all **maximal,unique** tuples in paragraph P.

3.3.3 Non-Unique Tuples

We have demonstrated how Algorithm 3.5 can identify all maximal, unique tuples in paragraph P. By *maximal* and *unique* we mean the largest non-overlapping tuples. Figure 3.11 shows two unique tuples. The first one, φ_1, has the n-gram overlap 'The deadline for', and the second one,φ_2, has the n-gram overlap 'You'. The second n-gram overlap is maximal in that it sacrifices overlap size in favour of matching a larger number of sentences. There are three other, non-maximal and non-unique tuple in the given paragraph. One such tuple is the one corresponding to the n-gram overlap 'You will', and it spans sentences three and five. It shares a partial sentence space with the tuple φ_2. We shall name it φ_x. The last two non-unique, non-maximal tuples are 'The', and 'The deadline'. Both these tuples have the same sentence span as tuple φ_1, but have a smaller n-gram overlap. These tuples will

You may want to pay the registration fee at the same time as you register. ***You*** *will* not be able to register over the weekend. ***You*** *will* be e-mailed your registration details upon payment receipt.

Figure 3.12: Two unique tuples in a paragraph.

not be generated by our algorithms, because they only provide partial information: Their n-gram overlaps are incomplete.

Tuple φ_x, however, is more interesting. Both tuples φ_2 and φ_x are complete, but they emphasise different goals. Tuple φ_2 focuses on tuple width, as currently implemented by Algorithm 3.5. When performing a comprehensive search for all possible tuples we are just as interested in the tuples of type φ_x as we are in the unique, maximal tuples like φ_2. In order to distinguish between maximal tuples and their variations we will take advantage of the recorded n-gram overlap size r from Algorithm 3.4. We have so far ignored this feature of our collection algorithm, but it gives us just the right information to discriminate between these tuples.

Algorithms 3.6 and 3.7 incorporate the use of the recorded n-gram overlap length r. The major change in Algorithm 3.6 is that it uses Algorithm 3.7 to generate non-maximal tuples from the unique maximal tuple *tuple*. Furthermore, we do not discard the entries that match the different S_b sentences. Instead, we use the information provided by them to determine the minimum size of the shared n-gram overlaps. The reason is that the shared n-gram overlap size of tuples of a particular size may be larger than those matched against the entries for the S_a sentences. To exemplify this, let us re-order the sentences from Figure 3.11 as shown in Figure 3.12. The maximum n-gram overlap for φ_1, as shown in **bold** text, is of size one. That is

Algorithm 3.6 Find the unique tuples in collection C

```
Require: $C = \rho_1, \ldots, \rho_n$
if $n < 2$ then
    return $C$ {There is only one tuple, a double}
end if
i:=1
while $S_{a,i} = S_{a,1}$ do
    $i := i + 1$
end while
$tuples := [\,]$
$tuple \leftarrow [\rho_1, \ldots, \rho_i]$
$remain = [\,]$
for $j := 1$ to $n$ do
    if $j > i$ then
        for $k := 1$ to $i$ do
            if $S_{a,j} = S_{b,k}$ or $S_{b,j} = S_{b,k}$ then
                $tuple \leftarrow \rho_j$
            else
                $remain \leftarrow \rho_j$
            end if
        end for
    end if
end for
$temp :=$ findNuTuples( $tuple$ )
for $m = 1$ to length of $temp$ do
    $tuples \leftarrow temp_m$
end for
$temp :=$ recurse( $remain$ )
for $m = 1$ to length of $temp$ do
    $tuples \leftarrow temp_m$
end for
return $tuples$
```

Algorithm 3.7 Find the non-unique tuples in subset Z

```
Require: $Z = \rho_1, \dots, \rho_n$
$maxng = 0$
for $i := 1$ to $n$ do
    if $r_i > maxng$ then
        $maxng = r_i$
    end if
end for
$nutuples = [\,]$
for $i := 1maxng$ to 1 do
    $temp = [\,]$
    for $j := 1$ to $n$ do
        if $r_{tuple,j} \geq maxng$ then
            $temp \leftarrow \rho tuple, j$
        end if
    end for
    $count = 0$
    for $j = 1$ to length of $temp$ do
        while $S_{temp,a,j} = S_{temp,a,1}$ do
            $count := count + 1$
        end while
    end for
    $minng =$ false
    for $j = 1$ to $count$ do
        if $r_{temp,j} = 1$ then
            $minng =$ true
        end if
    end for
    if $minng =$ true then
        $nutuples \leftarrow [\rho_{temp,1}, \dots, \rho_{temp,count}]$
    end if
end for
return $nutuples$
```

because the first sentence is only capable of matching one token of the other two sentences. If we rely our measurement of the non-unique tuples only on the length of the n-gram overlaps involving S_a, then we will only be able to locate tuples with an n-gram overlap of size one. If, however, we search the entries generated by matching the S_b sentences against each other, then we are able to locate the proper non-unique tuple φ_x with an n-gram overlap size of two (in *italics*).

3.3.4 Distribution of epanaphora by tuple size

When running our input documents through Algorithm 3.6 we expected a similar distribution of instances of epanaphora as we had found when performing the n-gram search. Indeed, the distribution in Figure 3.13 we see that this trend continues. The apparent tapering on the last column is due to it including tuples of size five or larger.

3.4 Summary

In this chapter we have defined the focus of our study, rhetorical anaphora. We have examined its common definition and produced our own disambiguated description. The purpose of this disambiguated description was to create a definition that could be easily applied to computational methods. This definition included the distinction between intentional and accidental epanaphora, the definition of primary attributes of epanaphora-like repetition, and an examination of their likely influence on the intentional nature of epanaphora.

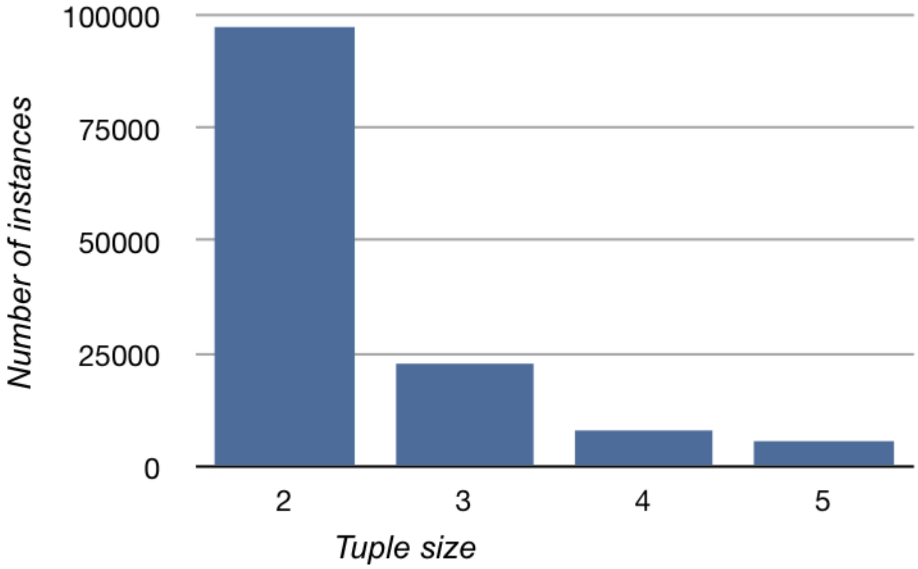


Figure 3.13: Epanaphora distribution by tuple size.

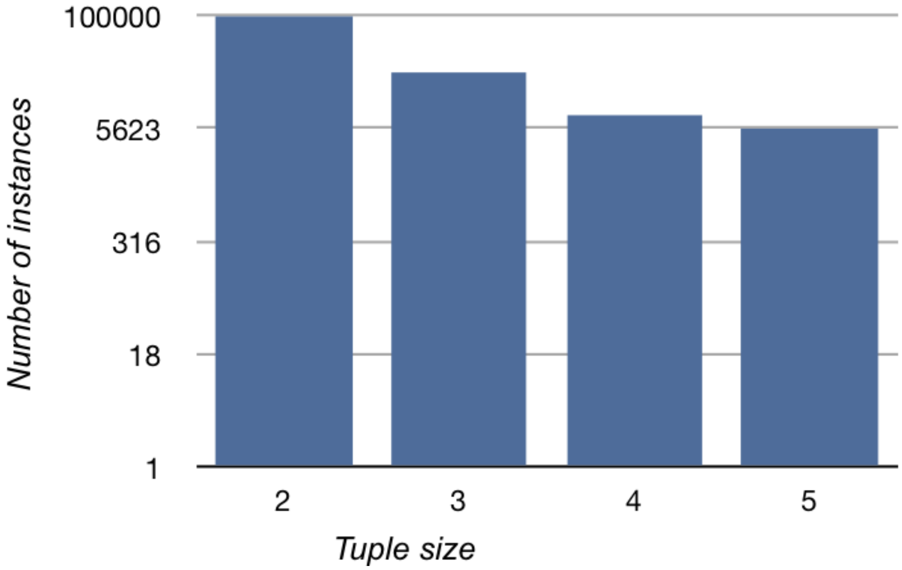


Figure 3.14: Epanaphora distribution by tuple size (logarithmic).

We also reviewed the process in which we selected a corpus for epanaphora detection and classification, and laid bare the justifications in favour of using the TREC Blog corpus.

Chapter 4

Epanaphora Detection

In this chapter we will examine additional refinements that may be layered on top of the two primary variations, n-grams and tuples. We call these refinements 'secondary' because they are not intended to extend the variation space of epanaphora beyond that of the primary criteria. Instead they are meant to aid in generating a more fine-grained method of differentiation between accidental, brute-intentional, and designed-intentional epanaphora. We will focus on three types of secondary variations: Gaps, keywords, and sentence length.

Once the secondary variations have been defined, we shall examine the methods that can be applied in creating an efficient epanaphora detection service. Beyond providing the benefit of performance gains, this section will also allow us to understand epanaphora better by helping us become familiar with their variation space.

Finally, we will perform a preliminary examination of the results of epanaphora detection, as well as study the steps necessary for generating a useful epanaphora

> I felt moody and irritable. I felt squished inside, I felt like standing in a field and twirling in circles with my arms spread wide, twirling and twirling and twirling until I fall. *Is it the driver's license?* I felt overwhelmed by it tonight.

Figure 4.1: Example of a gap between epanaphora constituents

classification service. In order to do so we will divide this step into three tasks: Early examination, prediction, and corroboration via a pilot study.

4.1 Gaps

In the context of epanaphora detection, gaps are any sentences (or phrases/lines, depending on the definition of epanaphora used) that are not part of a repetition, but occur between the constituents of said repetition. Figure 4.1 shows an example of a gap of length one (in *italics*). There are four sentences in that paragraph, but only three of them (sentences one, two, and four) contain the repetition *I felt*. Sentence three is an interruption of the repetition, sequence, a 'gap'.

4.1.1 Relevance

In comparing n-gram and tuple searches (Figure 3.10) it was stated that one major difference between them is that tuples allow gaps between their constituents. That aspect had not been examined in detail because it did not hold relevance to repetition of elements, the patterns of epanaphora classification studied in Chapter 3. However, closer examination of the impact of gaps on the quality of epanaphora promises to improve our knowledge on the use of intentional repetitions.

Just as it was possible to generate variations by putting boundaries on tuple and n-gram searches, so too is it possible to introduce another dimension of variation by placing limits on the size of the gaps between sentences forming a tuple.

The most salient reason for considering gaps between constituents of epanaphora is the particular wording of the definition of epanaphora.

> ***Epanaphora:*** *Repetition of the same word or group of words at the beginning of successive clauses, sentences, or lines.*

The key words that we will focus on in this section are *successive clauses*. In Chapter 3 we formed the definitions of the primary attributes of epanaphora based on the assumption that *successive clauses* implies sequence, and that the order of occurrence of the constituents of an epanaphora is of importance. However, *successive clauses* can also be interpreted as *consecutive clauses*. This second definition implies not only that the sentences that make up an instance of epanaphora are sequential, but also that they are in close proximity of each other. In its most strict sense, it implies that the only gap size that is acceptable is a null-gap. We will not adhere to this strictest definition of 'consecutive' because there are numerous cases where a gap may occur between the constituents of an epanaphora. An example of such has already been shown in Figure4.1. Gaps may be intentional by the author, or they may be the result of an error on the part of the sentence detector. These errors can occur in unsupervised sentence detection tools where there is an ambiguous use of punctuation, such as uncommon abbreviations and mixed descriptive text and dialogue.

Despite the having shown that naturally and unnaturally-occurring gaps between the constituents of epanaphora are possible, it can be contended that it is important to maintain a minimal distance between these constituents. The reason for such an argument is rooted in neuroscience and psychology, in particular the study of short-term memory, and is commonly referred to as the *memory span*. Memory span is the longest list of items that a person can repeat back in correct order. If the gap between two or more constituents is too large, then recollection of the earlier constituents can be hampered. As a consequence of the lower recollection of earlier constituents of repetition the saliency of said repetition, and therefore its rhetorical effect, are diminished. We therefore claim that larger gaps between the constituents of an instance of epanaphora can be linked to lower perceived intentionality of said instance by human readers. We follow up on the justification for lower perceived intentionality by readers and apply it to the author's perspective: An author that is aware of the theory of memory span will attempt to counteract the effect of large gaps on the recall of of his/her audience by actively avoiding placing the elements of a repetition too far apart.

We propose that intentional epanaphora encountered contain gaps shorter than the 'magical number seven' [103]. In its most rudimentary application this number will provide a reasonable initial threshold value for performing pruning of detection results, should the need arise. By starting out with this value as the threshold and examining its variation on the intentionality of detected instances of epanaphora we can furthermore provide additional circumstantial evidence to support or contradict the postulations of the 'magic number seven' theory and its derivatives.

4.1.2 Gap Detection

As mentioned in the introduction to this Chapter, we have decided to classify gaps as a secondary variation. Given the previous discussion on the relevance of the study of gaps in epanaphora, it can be argued that gaps should perhaps take a larger role in epanaphora detection, and that they should perhaps be classified as primary attributes of epanaphora. The chief reason for not doing so is that gaps do not extend the dimensions of variation of n-grams and tuples combined. In other words we mean that there is no sentence that can be selected via the addition of gap constraints that cannot be selected via the primary attributes. The simplest method of backing this statement is to go back to the proofs in Chapter 3. We demonstrated that the primary attributes perform a comprehensive search of the valid sentence variations for epanaphora. The way we use gaps as a secondary attribute of epanaphora in this thesis is to split any sequence of repeated sentences based on their proximity. Since the initial sequence has already been selected by primary attributes, and the sum of the sub-sequences cannot have more elements than the initial sequence we can say beyond shadow of a doubt that the application of the gaps constraints attribute is performed as a constrained mechanism of selection by proximity on top of n-gram length, and therefore is not capable of selecting any new sentences for repetition.

Both the gaps and the tuple width attributes are variations of the tuple space. Tuple width selects by frequency, gaps select by proximity. Given the choice of applying one attribute before the other we chose to give precedence to the tuple-width attributes over the tuple-gap attributes. As a result n-gram and tuple-width variations are applied first, and gap variations are applied as a filter on top of those.

Nevertheless, the resulting dimension of variation is equal to the one returned if n-gram and gap variations are applied first, then tuple-width. The reason for the results being the same is that tuple-width selection and tuple-gap selection are independent constraints which could be applied simultaneously. However, the advantage of performing a tiered system of attribute detection (as opposed to a parallel implementation) is that the additions can only limit the dimensions of variation, not expand them. Taking the n-gram space η, the tuple space φ, and the gap dimension γ we can write this relation as

$$0 \leq \eta \cdot \varphi \cdot \gamma \leq \eta \cdot \varphi \tag{4.1}$$

By simplifying this formula we get

$$0 \leq \gamma \leq 1 \tag{4.2}$$

Since γ is no longer unbound it should no longer have a significant bearing on the overall complexity of the tiered attribute detection system. Such a claim can be easily demonstrated by resorting to a property of the size of gaps, namely that the maximum number of gaps is equal to the maximum number of consecutive pairs within the constituents of an instance of epanaphora. Therefore if there are n constituents, then there will be a maximum of $n-1$ gaps. We can look at the sentences and gaps as an acyclic path graph. Each edge can have a weight ew_i where $i \in \mathbb{N}^0$. Edges of weight $ew_i > 0$ are counted as gaps. The edge weight is determined by the number of non-constituent sentences that occur in a paragraph between constituent sentences of epanaphora. Since each constituent sentence corresponds to one node and there are a maximum of n nodes, we have $n-1$ edges. The maximum number of gaps is $n-1$ when all edges have weight $ew_i > 0$. By tying the size of the γ dimension to that of φ space we can collapse the complexity formula:

$$O(\eta + \varphi + \gamma) = O(\eta + \varphi \cdot 2) = O(\eta + \varphi) \tag{4.3}$$

We have thus shown that the complexity of epanaphora detection with ngram-length, tuple-width and gap-length is no worse than that of ngram-length and tuple-width alone.

4.1.3 Implementation

The core implementation of detecting the gap size of a collection C of sentences in an instance of epanaphora revolves around comparing the indices of the constituents of said collection in their source paragraph P. The gap between two sentences S_a and S_b (occurring in that order) is achieved by subtracting the index of S_b from S_a and reducing the result by one. The last reduction is necessary because the index of consecutive sentences is incremented by one for each sentence. Since consecutive sentence are gapless, we need to adjust for that increment. This approach is embodied in Algorithm 4.1. Combination with n-gram detection follows the same steps as Algorithms 3.6 and 3.7.

Algorithm 4.1 Calculate the size of the gap between ordered sentences S_a and S_b

```
Require: $S_a, S_b$
return $index_{S_b} - index_{S_a} - 1$
```

Due to significant alterations to the design of the core algorithms later in this Chapter we will only keep this algorithm at the prototype stage until the new core of the epanaphora detector has been implemented.

4.2 Keywords

The type of words used in a repetition can have a significant impact on the effect of said repetition. We argue that certain words and classes of keywords have a strong relation to epanaphora, and that they can be used to disambiguate between intentional epanaphora and accidental repetition.

In this study we focus solely on the first word of each repetition. The reason for narrowing our keyword focus is that the relative placement of the keywords within a sentence is just as significant to the function of said keyword as the relative placement of repetitions with respect to each other. Since we can only guarantee the consistent placement of the first word of each repetition we decided to limit our keyword search to only those. A residual effect of this decision is that is becomes easier to select classes of keywords for separate classification.

Two primary categories of words are recorded: The direct word, and the keyword category. The direct word is the token as it appears in the sentence, and requires no training. The keyword categories are defined below.

We use a frequency-based selection method for keywords. This selection is performed iteratively by selecting a frequently-occurring keyword, finding a word class that can be used to generate more matches, and grouping all the results for that class. In some cases a chosen class becomes too large, a particular keyword has significantly higher incidence than the other words within that class, or a keyword is deemed to be used primarily for a different purpose than the rest of its class. In such cases a separate sub-class is created specifically for those words.

The rest of this section is dedicated to examining the particular classes of keywords that arose from the strategy described above. We will show the classes, which keywords they encompass, and what the major perceived purpose of each class is (based on a cursory examination of common results.

4.2.1 Personal Pronouns

> ***She*** *was disappointed.* ***She*** *could have but little idea that it was of firm purpose he avoided her.* ***She*** *could not know that he, like herself, had recognised the sympathetic mind, but, unlike herself, had recognised far more..*

The *personal pronoun* class of words encompasses nouns which act as a substitute for proper nouns and other nouns. What makes this class interesting is that it was identified before the keyword classification process was put in place. Personal pronouns, when used at the beginning of sentences, establish the identity to which the rest of the sentence is applied. This base identity is more commonly known as *identity of reference*. Establishing such an identity is a key step to successful persuasive oratory.

Another reason to look at personal pronouns in particular is because their nature as noun substitutes means that they are frequently used in linguistic anaphora (coreference). It might thus be plausible that we could find a correlation between the use of linguistic anaphora and the use of rhetorical anaphora.

Predictions

Our prognostic is that the use of repetitions of personal pronouns shall fall into two categories: The expression of a feeling or desire, and event recounts. Both categories can be used to create a strong rapport with the audience. The former is commonly used in a direct appeal to the audience's pathos by emphasising the humanity of the speaker and usually generates a positive emotional link between the orator and the audience. Event recounts, on the other hand, are an indirect appeal by engendering a feeling of community. This indirect appeal can be either towards the audience's positive emotions by defining the audience's community and situating the orator within that group, or it can appeal to the audience's negative emotions by playing on the audience's negative emotions to a third party.

Self Identity

> ***I** have never gazed on the lofty summits' of the Andes. **I** have never seen the Niagara. **I** have never roamed through classic Greece.*

One particular personal pronoun deserves its own category, and that is the self identity as per the first person singular '*I*'. The reason for this isolation of the first person singular case is the sheer number of unique instances of repetition that feature this keyword. The TREC '06 Blog corpus features these repetitions with unusually high frequency due to the nature of its contents.

In addition, first- and second-person personal pronouns are very rarely if at all used for linguistic anaphora. However, second-person and first-person plural pronouns are unlike the self identity in that they do not occur often enough in the given corpus to warrant their own sub-category.

4.2.2 Question Keywords

> ***What*** *is the use?* ***What*** *is the use of all these vain efforts?* ***What*** *is the use of all these monotonous beginnings?* ***What*** *is the use of playing so burdensome a part upon the social stage?*

Questions were the first category of repetitions to be chosen with the frequency-based selection method. We decided to limit this class of keywords to what is known as '*interrogative words*'. These words contain a sub-class commonly referred to as the '*five Ws*': 'who', ''what', 'when', why', 'where', plus 'how'. The vast majority of sentences beginning with interrogative words use the six keywords cited above. We speculate that the reason for the focus on these particular interrogative words is that the other words in the interrogative class have been deprecated from colloquial English.

It is possible for question sentences to begin with keywords other than interrogative words. However, we decided to limit the selection of keywords for the question class to interrogative words. The reason for limiting our system to these keywords is purely pragmatic. The set of interrogative words has a fixed bound, making detection a brisk task. Question sentences that fall outside this group however are more difficult to classify. In order to minimise the classification error it was thus necessary to exclude non-interrogative words from the question keyword class.

Predictions

Given the medium from which our corpus was extracted we predict that a large portion of the repetitions that contain question keywords *and* are intentional epanaphora will be rhetorical questions. The justification for this prediction is that rhetorical questions have the strongest incentive for repetition, whereas questions which focus on information extraction tend to avoid redundancy.

Special Case: ‘When’

> ***When*** *shall my feet, on earth so tired that grew,*
> *Tread all unfalteringly the street of gold?.*
> ***When*** *shall I come where trees of healing bend*
> *Their deathless boughs the living stream above?*
> ***When*** *shall I listen to the music sweet*
> *That thrills the glad air of the land I love?*

After re-examining the results for the question keyword class we discovered that one particular case was different from the rest, namely the keyword '*when*'. As with the self identity in the personal pronoun class, the particular corpus that we chose influences the use of this word. Beyond its use in questions, '*when*' can also be used for recounts of events. We decided that, just as was done with the self identity, it is warranted to create a separate sub-category for this keyword.

4.2.3 Conjunctions

> *My mother actually took me shopping today! I got two skirts.* ***And*** *i got two tops.* ***And*** *i got some shoes and jewelry.*

Conjunctions are connector-type words that are used to bind adjacent phrases or clauses. These conjunctions are not intended to be used at the beginning of sentences, and any such application of these parts of speech is considered to be grammatically incorrect. The fact is nevertheless that such sentences are encountered with unusually high frequency in our corpus. The reason for this occurrence is that many of the examined blog entries straddle the borderline between prose and spoken dialogue. Recounts of events in particular tend to dominate this category.

Predictions

We don't expect this category to produce many *sentence*-type intentional repetitions. We believe that repetitions of conjunctions at the beginning of sentences may be good indicators for *phrase*-type repetitions. Furthermore we will argue that this type of repetition will not be able to match complete intentional repetitions because they are not able to match the initial phrase.

While detecting conjunctions at the beginning of sentences may produce good intentional repetitions, we argue that results matching these parameters should be excluded from future parses due to them being *phrase*-type repetitions as opposed to *sentence*-type repetitions.

4.2.4 Articles

> ***The*** *CFD also asked for the removal of the movie Splash.* ***The*** *CFD declared war with pickets and boycotts. (. . .)* ***The*** *situation worsened when the couple received death threats to their little daughter.*

Articles are parts of speech used to define the type of reference made to the noun they are attached to. Modern English has two sub-classes of articles: Definite and indefinite. The former consists of the single word '*the*', and is used to indicate that the noun it is attached to is a specific one which the audience can identify. '*A*' and '*an*' form the indefinite article class, and are attached to nouns which are not uniquely identifiable, such as general categories.

Predictions

Articles are extremely common at the beginning of sentences, and as a result they are very likely to form repetitions throughout a paragraph. Their commonality also dilutes their rhetorical strength. Due to this we predict that the repetitions generated from them will be largely of an accidental nature.

4.2.5 Conditionals

> ***If*** *the frame is flexible enough to match your building, you will do well.* ***If*** *the frame does not match, you may be worse off than if you had no frame at all.* ***If*** *the frame is inflexible, you will see builders abandon the frame and build their own.*

There are two types of condition that we will focus on: Branches and iterations. Branches are sentences that begin with the words *if* and *unless*, whereas iterations begin with the words *while* and *unless*. Branching keywords are somewhat frequent at the beginning of sentences. Iteration keywords, in contrast, are relatively rare.

It can be argued that branches and iterations should be in separate categories. However, examination of results shows that there is little impact in keeping these two types of keywords under the same category.

Predictions

In colloquial English, repetitions beginning with branch conditionals are most commonly used in rhetorical figures related to contrast, such as antithesis. As a result of their nature, contrast is the primary goal of these instances, not repetition. However, in many cases there will be a duality in effect, where repetition takes either a secondary role, or where contrast and repetition are co-primary purposes. Furthermore we predict that the particular choice of repeating the branching keyword during sequential sentences constitutes a wilful decision to emphasise the repetition aspect, and that most of the found instances shall be classified as intentional epanaphora.

4.2.6 Demonstratives

> ***This*** *day I baptized Marie Joseph, recently born of the marriage of Franois Le Beau and Marie Josephine Bigra.* ***This*** *day I baptized Michel, recently born of the marriage of Charles Buteau and Marie Marguerite Gautier.* ***This*** *day I Supplied the rites of baptism to Jean, recently born*

> *of the marriage of Jean Brisar and Marie Angelique Clement.*

Demonstratives are words which are used to distinguish between objects of entities being referred to by a speaker. They are linked to deixis, the need for contextual information to understand a word or phrase. In the English language there are two types of demonstratives: *proximal* and *distal*, indicating the relative distance between the entities and the speaker. The keywords that are used for the demonstrative filter are *this, that, these*, and *those*.

4.2.7 Possessives

> ***My*** *times are in Thy hand; My God!* ***My*** *times are in Thy hand, Whatever they may be.* ***My*** *times are in Thy hand; Why should I doubt or fear?*

In the English language there are two types of parts of speech indicating possession: Pronouns and adjectives. Possessive pronouns are seldom used at the beginning of sentences in colloquial English. However, possessive adjectives occur frequently in our corpus. There are seven possessive adjectives, derived from personal pronouns: *My, your, his, her, his, its, our, your* (pl.), and *their*. They are used to ascribe ownership of a noun to a subject. For the purpose of generating a keyword filter we are not required to distinguish between the singular and plural forms of *your*.

Predictions

We estimate that repetitions starting with possessive adjectives will have a good likelihood of being intentional epanaphora. The justification for this prediction is that the expression of ownership is attached to strong emotional expressions. Furthermore, being parts of speech that are derived from personal pronouns we expect that they will share a similar rhetorical influence. As a result we expect that the distribution of epanaphora among possessive adjectives will mirror that of personal pronouns.

4.3 Design and Performance

Performance design is an often-overlooked aspect of computational linguistics research. The potential runtime improvements alone would warrant a second look at attempting to implement a fast epanaphora detection algorithm, particularly if the design allows the system to perform real-time epanaphora detection and classification.

Revising the epanaphora detection algorithms also brings along benefits from a theoretical perspective. By studying the performance of the detection process we are able to garner a better understanding of the mechanics of repetitions within the target corpus.

Finally, a redesigned algorithm will allow us to introduce options for more fine-grained epanaphora detection and classification.

4.3.1 Epanaphora Revisited

The first task before engaging in a core redesign of the epanaphora detection process is to re-examine what we know about both epanaphora and the detection process. A fresh look at both the composition of the rhetorical figures as well as the nature of the epanaphora detection process will enable us to proceed in a structured manner that is more suitable to gathering and laying out the information that is necessary to improve the detection process' performance.

Epanaphora Traits

Epanaphora is, by definition, a form of repetition. Furthermore, we know that it has a strict left boundary, namely the beginning of a sentence. This boundary was imposed by our own design as we chose to focus on repetitions across sentences as opposed to repetitions across sentences, phrases, and lines of text. Knowing that our repetition searches all begin at the same token within each sentence gives us the first clue as to how we will improve our epanaphora detection algorithm.

Of all the attributes we identified for detecting and categorising repetitions, the n-gram length is the strictest. The n-grams used have a fixed left boundary and a right boundary limited at least by sentence length. We also know that comparing n-grams is a sequential process and does not allow gaps. Those features of n-grams are further clues for improving the epanaphora detection algorithm. The left boundary for n-grams is the same as the left boundary for epanaphora. The right boundary at sentence-length tells us that the longest n-gram including every sentence in set in a way that S can be no longer than the shortest sentence S_i where $1 \leq i \leq n$ and Equation 4.4 applies.

$$\forall(S_k) \bullet k = \{1, \ldots, i-1, i+1, \ldots, n\} \wedge length_{S_k} \geq length_{S_i} \tag{4.4}$$

Still, we can improve – shorten – the boundaries with the inclusion of the additional features of n-grams, namely their lack of gaps. A lack of gaps indicates that the moment a token in any of the examined sentences stops matching the rest of the patterns that sentence is no longer a candidate for longer n-gram matches. We can use this knowledge as a constraint to discard sentences that do not match a minimum n-gram length. Furthermore, the same method can be reused to create additional boundaries for tuple and gap detection.

4.3.2 Comprehensive Epanaphora Detection

In Chapter 3 we built our epanaphora detection algorithms around the concept that only a particular range of the included features determines the quality of an instance of epanaphora. While we still hold this idea to be true, the number of combinations available has increased drastically due to the introduction of additional classification criteria. As a result it is no longer practical to target specific combinations. More importantly, during the pilot we have shown that the initial speculation about the role of certain detection and classification criteria may have been incorrect. It is therefore reasonable to perform the first classification round on all ranges for each of the classification criteria. Since the initial design of the epanaphora detection algorithm was not built for this goal we will dedicate this section to generating a new epanaphora detection algorithm that is both flexible and efficient.

As already mentioned, the previous goal of the epanaphora detection system was to find repetitions matching an open range for each attribute. Our aim with the new system it to implement mass-detection, whereby the goal is to find every unique and atomic variation. In other words we want to set the attribute range to always be minimal for each result. While our current method can be adapted and re-used by setting the condition for each attribute boundaries as narrow as possible (i.e. setting the lower bound to the same value as the upper bound), using that approach means increasing the running time by at least a polynomial degree, since the system requires a full iteration over the entire corpus for each unique attribute combination. Furthermore, since the new approach is intended to permanently use the same value for the upper and lower bounds it is no longer necessary to distinguish between them.

Our primary goal in the following sections will be to avoid repeated full iterations, as well as minimising the work that is needed to be performed when certain attributes are not altered over different local iterations. The most direct approach to implement the latter is to begin with a divide-and-conquer approach to reduce the range of values which affect the polynomial complexity of attribute combinations. Once that is done we shall introduce caching of intermediate results. Our task there will be to identify those intermediate steps and select the ones that are most appropriate for performance improvements terms of the information recorded, the possibility of minimising memory usage, as well as the effect on run-time complexity due to their introduction. In addition to the performance gains attained by the introduction of caching methods, we will also gain a better understanding of how different types of epanaphora share common traits.

4.3.3 Improvements

Our first approach to finding information that can be cached will center around a divide-and-conquer method. We will identify clusters of information that can be examined independently. Even though this method alone will only produce a constant-rate improvement, it is based on one of the most fundamental facts about epanaphora detection.

Clustering

We know that the absolute left boundary of our target repetitions is the beginning of each sentence examined. We furthermore know that in order for two or more sentences to form a repetition pattern they need to share at least the very first word of each sentence. By combining these two pieces of information we can take all the sentences in a paragraph and sort them by their first word. We thus have split the paragraph set of sentences S into subsets $\zeta_1, \ldots, \zeta_n$. Each subset is completely independent of each other, and we are guaranteed that there is absolutely no chance of a repeating pattern to occur across two different sets. Doing so would violate the requirements set in the definition of epanaphora n-grams, where they *must* start at the beginning of a sentence and they *must not* skip any word. An additional side effect is that we can easily classify any resulting repetitions by keyword - instead of performing keyword classification on each sentence in a paragraph we only need to run the keyword classifier against each of the sentence sets. As a result we have implemented our first real performance improvement by collapsing the execution of a linear classifier to constant time.

A more important side effect of the above approach is that we have not only performed keyword classification, but that the clusters themselves are results already. Each subset $\zeta_1, \ldots, \zeta_n$ contains the most general result for each sentence-starting word. We know that there cannot be any more sentences that might match that word, because doing so would mean that the initial cluster subset division would not complete at that point.

Algorithm 4.2 shows a possible implementation of the above design. It demonstrates the simplicity of the clustering approach and showcases how it can be performed in a single iteration over the collection of sentences. . Clustering begins on Line 3 and ends on line 7. Furthermore the sequential nature of this approach means that by the time the clustering ends each array in the *clusters* hash contains the sentences in the same order in which they were encountered in the original sentence array S.

Algorithm 4.2 Perform keyword-based clustering of sentence collection S

```
Require: $S = S_1, \ldots, S_n$
 1: $clusters$ := new Hash( key:String, value:[ ] )
 2: $i := 1$
 3: while $i <= n$ do
 4:    $kw := S_i[1]$ {Retrieve the first word $kw$ for sentence $S_i$}
 5:    $clusters[kw] \leftarrow S_i$ {Append $S_i$ to the hash array indexed by its first word}
 6:    $i := i + 1$
 7: end while
 8: for each array $Cl_i$ in $clusters$ do
 9:    perform epanaphora detection on $Cl_i$
10: end for
```

Attribute Recording

Having identified the general clusters of sentences $Cl_1 \ldots CL_n$ we can proceed to record the attributes of the epanaphora variations present in each of them. In Section 4.3.3 we postulated that the identified clusters are themselves results. We will first expand on that.

The cluster subdivision is performed using the first word of each sentence as the clustering criterion. Clustering sentences by their first token guarantees the following:

Observation 4.3.1. *All the sentences within one cluster share the same word at their beginning.*

In addition, we can assure that

Observation 4.3.2. *No sentence in any other cluster contains the same first word.*

Observation 4.3.1 is self-evident from the description of the clustering method. Observation 4.3.2 follows from Observation 4.3.1. If there were to exist two clusters Cl_a and Cl_b such that at least one sentence from Cl_b shares the same keyword as at least one sentence from Cl_a, then by Observation 4.3.1 we know that *all* sentences from Cl_a and Cl_b share the same initial word. Two separate clusters sharing the same token is evidently a contradiction to the clustering design. Observation 4.3.2 must therefore hold true. Knowing that every sentence within any cluster Cl_i gives us an additional benefit, namely that we do not need to record the keyword attribute for each individual sentence within said cluster. Because we ensured that all sentences within Cl_i share the same first word and then used said word as the hash key for the cluster collection in Algorithm 4.2 we can use said key to set the keyword attribute for all epanaphora variations which are based on the sentence cluster paired with that key.

We can now extend the knowledge that each cluster contains sentences that begin with an unique word. The basic rule for our epanaphora candidates is that each sentence in an instance of an epanaphora candidate must begin with the same word or words. By using Observation 4.3.1 we know that all sentences within any sentence cluster Cl match this rule. In addition, the introduction of Observation 4.3.2 tells us that the number of sentences in Cl is the maximum number of sentences (out of those that were given as input to Algorithm 4.2) that contain the same initial word as any of the sentences in Cl. Since the width of an epanaphora instance is equal to the number of sentences that match its other criteria we can further determine that the maximum with of any epanaphora instance for any keyword k is no greater than the number of sentences in the cluster Cl_k for such keyword k. In addition we know that only adjacent sentences within each sentence cluster Cl_i need to be compared. Even if there were instances of epanaphora with n-gram length longer than one – In other words, instances of epanaphora which are based on keyword matches beyond each sentence's first word – then those instances would still be no wider than the number of sentences in Cl_i, or they would violate Observations 4.3.1 and 4.3.2.

At this stage it is known that no epanaphora instance for any keyword k has greater width than the respective cluster Cl_k. We also know that there is no epanaphora candidate with shorter n-gram length than one, and that the cluster Cl_k has a guaranteed length of at least one. We further postulate that there is no epanaphora candidate E_k such that E_k and Cl_k share the same initial sentence word and where the n-gram length of E_k is less than that of Cl_k.

Observation 4.3.3. *For any sentence cluster Cl_k and key k there is no epanaphora candidate E_k with an n-gram length shorter than that of Cl_k.*

We know that Cl_k contains all sentences beginning with k. As a result, we know that any epanaphora candidate E_k must be a product of sentences from Cl_k. What is left to demonstrate is that there is no permutation of a subset of sentences from Cl_k which produces shorter n-gram length than the entire set of sentences from Cl_k itself. But first we shall revisit the n-gram length recording process itself. We only need to measure the n-gram length for adjacent sentence pairs S_i and S_j where $j = i + 1$.

Hypothesis 4.3.4. *For any candidate epanaphora instance E_k the n-gram length of said candidate is equal to the shortest n-gram length among any two pairs of sentences S_i and S_j from E_k where $j = i + 1$.*

To back Hypothesis 4.3.4 we need to show two things: First, that only one sentence is necessary to influence the n-gram length of the entire set of sentences in E_k and second, that any two adjacent pairs involving said sentence are enough to set the lower bound for this n-gram length.

Observation 4.3.5. *For any epanaphora candidate E there is at least one sentence which determines the lower bound for the n-gram length among the entire set of sentences within E.*

Observation 4.3.6. *It is sufficient to observe adjacent pairs of sentences to find one sentence which gives the lower bound n-gram for E.*

First we shall address Observation 4.3.5. We will begin with a comprehensive search among all possible sentence pairs within the given epanaphora candidate E. The base case is where all sentence pairs produce the same n-gram length. If that is the case, then any one sentence pair will produce the shortest n-gram length, which in this case is also the longest n-gram length.

The step is to introduce one sentence pair that has different n-gram length than the rest. We begin by introducing one pair of sentences which produces a larger n-gram length among the two sentences involved than any other pair. The remainder will continue to produce the shorter n-gram length among themselves, but also when pairing any of the short-n-gram sentences against the long-n-gram sentences. Following that, we proceed to replace each of the shorter n-gram sentences in E with one that matches the longer n-grams. As before, any of the sentences involving the shorter pairs is sufficient to set the lower bound for the n-gram length of E. We continue with this process until only one sentence remains which produces shorter n-gram length against the rest of the sentences. This last sentence is now the lower-bound determining sentence from Observation 4.3.5.

An important variation is when a sentence S_j is added to produce a longer n-gram pair against the lower-bound determining sentence S_i from Observation 4.3.5. Even though these two sentences produce a pair with a larger n-gram length than the lower bound, the original lower-bound sentence still produces the same bound length against all the other sentences. In addition, sentence S_j of the new pair also produces the same lower bound as S_i.

Observation 4.3.7. *Given any two sets of sentences ζ_a and ζ_b for which one sentence* *of ζ_a produces a shortest n-gram length against any sentence for ζ_b, then any other sentence from ζ_a also produces the same shortest n-gram against any sentences from ζ_b.*

The property described in Observation 4.3.7 can be extended to any number of sentence groups, and the order of the constituent sentences for each group has no effect on the final result. These attributes are crucial for Observation 4.3.6. They indicate that in order to determine the shortest n-gram length it is only necessary to find any transition between ζ_a and ζ_b. in the worst-case scenario this transition is where the sentence groups are not intertwined. In such case the only times where the n-gram length among sentences differs is at the boundaries between sentence groups. Still, whether the sentence groups overlap or not, a linear pairwise search will find at least one transition that generates the shortest n-gram length for epanaphora candidate E.

The last attribute that needs to be recorded for any sentence cluster Cl is the gap between the sentences within the cluster. However Algorithm 4.2 does not record the absolute position of the sentences within the input sentence set, only the relative order among each of the sentences. Therefore the first thing that needs to be done is to modify the algorithm to provide the necessary information.

The feature that Algorithm 4.3 introduces is the recording of each sentence position alongside the sentence itself on line 5. The information on the sentence positions can be used in conjunction with Algorithm 4.1 to calculate the gaps between adjacent sentences within each sentence cluster Cl. However, unlike the keyword, tuple width, and n-gram length there is not a guaranteed fixed gap width across each sen-

Algorithm 4.3 Perform keyword-based clustering of sentence collection S

```
Require: $S = S_1, \ldots, S_n$
$clusters$ := new Hash( key:String, value:[ ] )
$i := 1$
while $i <= n$ do
    $kw := S_i[1]$
    $clusters[kw] \leftarrow [S_i, i]$
    $i := i + 1$
end while
for each array $Cl_i$ in $clusters$ do
    perform epanaphora detection on $Cl_i$
end for
```

tence cluster. It is therefore necessary to record the necessary variations of tuple gap for each sentence cluster. For the purpose of this study those variations are the minimum, maximum, average, and median gap values.

Algorithm 4.4 implements the entire attribute recording process.

Recursion

Once all attributes of the base case have been recorded we proceed with finding narrower results. The first step to narrowing the results is done by reducing each sentence cluster Ci_i. The reduction is achieved by generating sentence subsets ζ^i out of every cluster Cl_i. One criteria for generating a subset is to cluster sentence pairs that generate an n-gram pattern larger than the lower bound found for Cl_i. In other words, the sentence subset from Observation 4.3.7 which is generating the shortest n-gram pair is separated from the other sentences, and each subset is treated individually as a disjoined cluster. Each of these subsets or disjoined clusters is a new result with an n-gram length larger than its parent. The separation and re-treatment

Algorithm 4.4 Perform attribute recording on a sentence cluster Cl

```
Require: $Cl = [S_1, j], \ldots, [S_n, k]$
$ngram\_length := 0$
$tuple\_width := n$
$tuple\_gaps := [\ ]$
$i := 1$
while $i < n$ do
    $S_a := Cl[i][0]$
    $S_b := Cl[i+1][0]$ {Fetch the sentences}
    $p_a := Cl[i][1]$
    $p_b := Cl[i+1][1]$ {Fetch the sentence positions}
    $\eta_{a,b} :=$ n-gram length between $S_a$ and $S_b$
    if $ngram\_length = 0$ or $ngram\_length > \eta_{a,b}$ then
        $ngram\_length := \eta_{a,b}$
    end if
    $tuple\_gaps \leftarrow [p_b - p_a - 1]$
    $i := i + 1$
end while
record attribute information for $Cl$
recurse on $Cl$
```

process is repeated until no more sub-subsets are generated.

The step described above deals with n-gram variations. Tuple variations are automatically taken care of because they are a simple by-product of the n-gram variations above. It is possible to generate narrower tuple patterns by eliminating sentences from the subsets and sub-subsets, but there is no practical goal in doing so, since no significant information is added. The very same result can be achieved in post-processing by performing a combination search.

A second variation process is possible, however, and that is gap variation. The method to achieve gap variation is similar to n-gram variations. However, order matters in this case, and we are interested in narrowing the gap between sentences, not increasing it. What we will do thus is to find the longest gap or gaps within each cluster Cl_i or cluster subset ζ^i and split them along those gaps. If the number of sentences within each subset is two or more then we can recursively apply the attribute-recording process on that subset.

The gap variation process is performed in parallel to the n-gram recording method, so duplicate results are possible. Dealing with these duplicates involves finding results where the sentence position numbers are the same as those of the current cluster. However, it remains questionable whether inline duplicate removal provides any performance improvement. The reason for such doubt is that the recursive processing of larger n-gram length and shorter gap width bounds does not produce an ordered set of results. What this means for inline duplicate removal is that it is necessary to perform a linear search in order to determine whether the current element is a duplicate of a previous result. The complexity of adding a linear search may be offset

Algorithm 4.5 Enter n-gram recursion on a sentence cluster Cl

```
Require: $Cl = [S_1, j], \ldots, [S_n, k]$
Require: $\eta_{min}$ = shortest n-gram in $Cl$
$Cl_{recursion} := [\ ]$
$Cl_{remainder} := [\ ]$
$i := 1$
$Cl_{recursion} \leftarrow [Cl[1][0], Cl[1][1]]$
while $i < n$ do
    $S_a := Cl[i][0]$
    $p_a := Cl[i][1]$
    $j := i + 1$
    while $j \leq n$ do
        $S_b := Cl[j][0]$
        $p_b := Cl[j][1]$
        $\eta_{a,b}$ := n-gram length between $S_a$ and $S_b$
        if $\eta_{a,b} = \eta_{min}$ then
            $Cl_{remainder} \leftarrow [S_b, p_b]$
            $j = j + 1$
        else
            $Cl_{recursion} \leftarrow [S_b, p_b]$
            break
        end if
    end while
    $i := j$
end while
record attribute information for $Cl$
run Algorithm 4.4 on $Cl_{recursion}$
run Algorithms 4.5 and 4.6 on $Cl_{remainder}$
```

Algorithm 4.6 Enter gap recursion on a sentence cluster Cl

```
**Require:** $Cl = [S_1, j], \ldots, [S_n, k]$
**Require:** $tuple\_gaps = [g_1], \ldots, [g_n - 1]$
$Cl_{recursion} := [\ ]$
$Cl_{remainder} := [\ ]$
$gaps_{remainder} := [\ ]$
$g_{max}$ := largest gap of $tuple\_gaps$
$Cl_{recursion} \leftarrow [Cl[1][0], Cl[1][1]]$
$i := 2$
**while** $i \leq n$ **do**
    $S_a := Cl[i][0]$
    $p_a := Cl[i][1]$
    **if** $tuple\_gaps[i - 1] = g_{max}$ **then**
        $j := i$
        **while** $j \leq n$ **do**
            $Cl_{remainder} \leftarrow [Cl[j][0], Cl[j][1]]$
            $gaps_{remainder} \leftarrow tuple\_gaps[j - 1]$
        **end while**
        **break**
    **else**
        $Cl_{recursion} \leftarrow [S_a, p_a]$
    **end if**
    $i = i + 1$
**end while**
record attribute information for $Cl$
run Algorithm 4.4 on $Cl_{recursion}$
run Algorithms 4.5 and 4.6 on $Cl_{remainder}$
```

by utilising a hash instead of an array to store the results, but it is unknown how often duplicate results are generated, and how much of an improvement the inline duplicate removal is versus a post-processing approach.

Chapter 5

Epanaphora Classification

5.1 Corpus Generation and Classifier Training

In this section we will cover two aspects of epanaphora classification. The first aspect is the generation of corpora for training, testing, and verification. The other facet is the use of those corpora to select, train, and evaluate candidate algorithms for epanaphora classification.

5.1.1 Annotation

The purpose of annotation is to create a baseline for training the epanaphora classifiers. This baseline is defined by creating a collection of instances of epanaphora that are representative of the 'intentional' set of epanaphora. The result of this baseline creation is that the goal of the first generation of automated epanaphora classifiers is to detect all epanaphora-type repetitions, excluding the ones considered to be accidental. To put this in a different perspective, we aim to find the types of repetition which generate audience interest towards the content.

Input

The input of the classifier training tool is the same as the output of the epanaphora detection algorithm. In this case that consists of a collection of documents in XML format. Each entry within the documents corresponds to the potential instances of epanaphora encountered during the detection process. The recorded information for each entry contains the paragraph in which the repetition was encountered as an ordered collection of sentences. The sentences which were identified as the constituents of the detected epanaphora are tagged by setting an element attribute. In addition to the paragraph we also recorded the attributes of each instance of epanaphora. These attributes are not used for the training process, but will be stored for later use by the classifier algorithms.

Output

In terms of format, the output follows the same structure as the input. The primary difference – in fact, the only difference – is that the output is sorted into separate document sets based on the annotations assigned to each paragraph. While this method fractures the data sets, it was decided to favour it over the addition of new XML elements and attributes. The reasons for this decision were ease of implementation and, more importantly, maintaining the homogeneity of the data structures. By using one format and sticking to it we can guarantee that the inputs at different stages of the classification process are interchangeable, enabling us for example to perform recursive annotation and classification. A further benefit to the approach described above is that by maintaining the cohesion between input and output the same annotation tool can be used to judge the quality of automated epanaphora classifiers.

Annotation

Under ideal circumstances, the goal of the annotation process is to separate the input data into two sets: Instances of epanaphora that are relevant to the goals of the annotator, and instances which are not. The instances which are relevant to the annotator can be tuned by changing the criteria of relevance that are given to the person. This annotation process is designed to minimise annotator error by reducing the number of choices given.

Interface

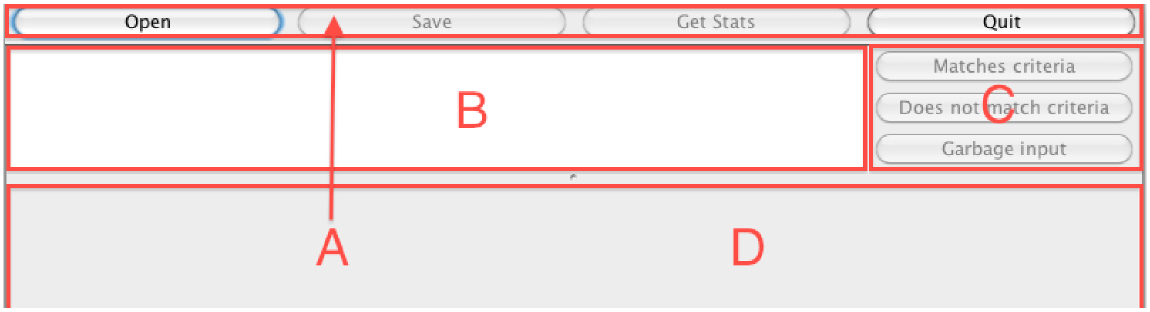
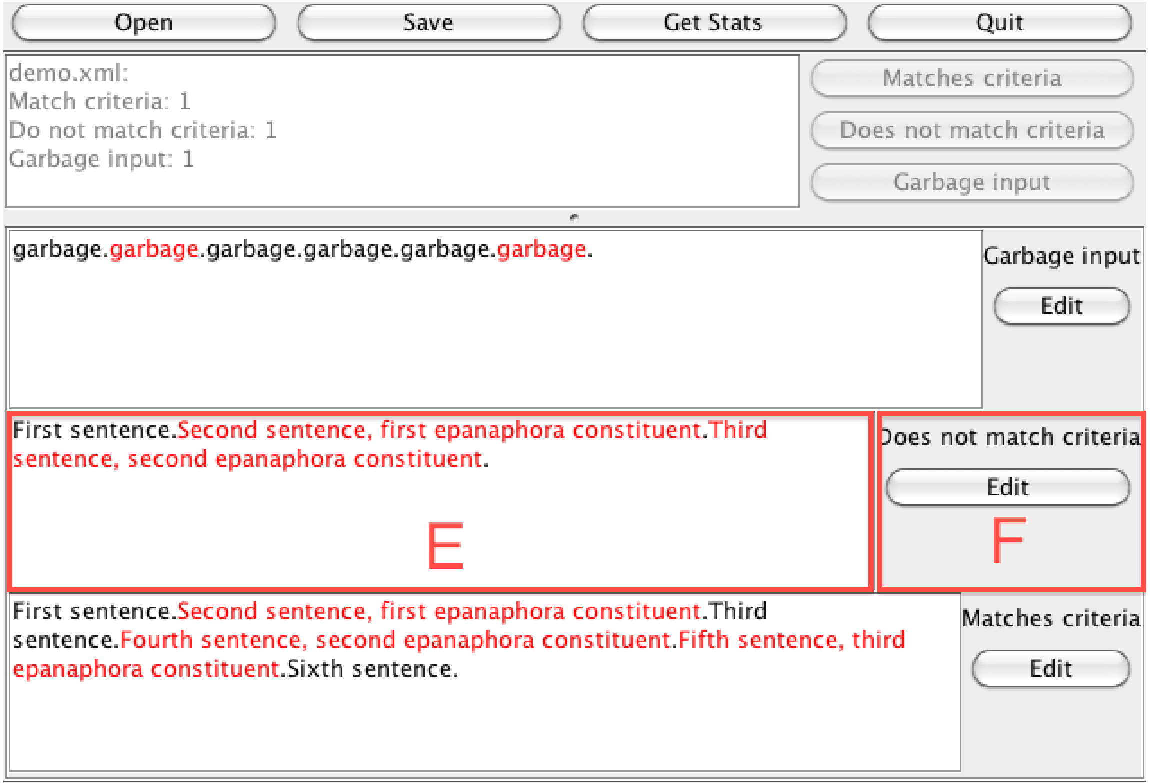


Figure 5.1: General view of the annotation tool interface

The design of the interface for the annotation tool follows minimalistic design patterns. It is composed of three major sections, which are shown in Figure 5.1. The first section is composed of area A. It is a panel that contains the general commands used by annotators: Opening input files, saving output documents, viewing annotation statistics for the current output, and closing the application. The second section is the primary annotation interface. Area B displays the instance of epanaphora to be annotated, and area C is the interface for the annotation selections described earlier in this section. The last element of the interface is area D, which is the review panel. It contains all the instances of epanaphora that have already been annotated for the current input file, displaying them in reverse order (newest at the top). The reverse order can be seen in the example Figure 5.2. In that example, the order of annotation was ‘matches criteria’, ‘does not match criteria’, and ‘garbage input’.

Figure 5.2 also displays further properties of the annotation tool. First of all, the review panel lets annotators compare their current annotations to previous ones. Being able to refer to their previous selections can be used by annotators to maintain consistency throughout the annotation process by generating a reference catalog of past annotations. The second property is that each element in the review panel is composed of two parts: The content display (area E), and the review controls (area F). The content display follows the same format as area B from Figure 5.1. In particular, it demonstrates that the application is capable of highlighting the repeating elements of an instance of epanaphora within each paragraph context. The highlights aid the performance of the annotators by creating a visual focus without the need to discard extraneous content. The second component of the elements in the review panel is used by annotators to alter the annotation choices for an element in



```
demo.xml:
Match criteria: 1
Do not match criteria: 1
Garbage input: 1
```

garbage.garbage.garbage.garbage.garbage.garbage.

Garbage input

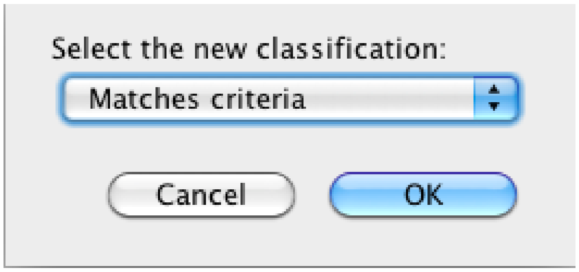
First sentence.Second sentence, first epanaphora constituent.Third sentence, second epanaphora constituent.

Does not match criteria

First sentence.Second sentence, first epanaphora constituent.Third sentence.Fourth sentence, second epanaphora constituent.Fifth sentence, third epanaphora constituent.Sixth sentence.

Matches criteria

Figure 5.2: Sample of the annotation tool review interface



Select the new classification: Matches criteria

Figure 5.3: Annotation review controls

the review panel. It provides access to the review controls shown in Figure 5.3. The primary goal of providing annotators with this option is to enable them to perform revision of their annotation choices after erroneous input. Such erroneous input is possible for example if an input document favours one of the classification choices. The repetitive nature of the annotation process is likely to lead to an attempt at faster element processing. There is a threshold however at which the repeated selection of the same choice becomes a semi-automatic, mechanical movement. Since the act of reading and comprehending each element of the input data is not instantaneous it becomes possible for this mechanical movement to trigger before or at the same time as a decision regarding the classification choice is made, thus resulting in input error. We decided however that the review controls should not be too accessible in order to prevent accidental alterations to the annotation choices. As a result we separated the review controls from the primary annotation tool interface and introduced a confirmation dialog. The confirmation dialog controls whether the changes are propagated back to the main interface, or whether they are discarded.

The last property that Figure 5.2 demonstrates is the annotation statistics review. It was introduced to provide an overview of the ratio of instances of epanaphora that match annotation criteria to the ones that do not.

Annotators

To perform the initial training of the epanaphora classifiers we hired annotators with expert knowledge of rhetoric. Each annotator is given a walkthrough to the annotation tool interface and a document describing the criteria for categorising instances of epanaphora, in human terms. Full disclosure was given as to the purpose of the annotation process.

Annotation Procedure

As was mentioned earlier in this section, the tasks given to the annotators are aimed at detecting sets of intentional epanaphora. However, we are also interested in measuring the quality of the datasets used to train the classification algorithms. It is therefore our intent to shape the annotation process in such a manner that will allow us to generate a fair balance between generating sets for the testing of inter-annotator agreement and the production of a dataset of sufficiently large proportions. The urgency of such a balance is compounded by the limited resources available during the annotation phase. In the end it was decided that the annotators would receive four sets of input data each. One of those four datasets is shared among all annotators, meaning that said set will be annotated independently by each annotator. The other three datasets are unique to each annotator. The shared dataset is reserved for measuring inter-annotator agreement. This reservation was done to prevent unbalancing the training process of the automatic classifiers. If the shared dataset had been included with the rest as input for the trainers then it could have had an unbalancing effect on the classification algorithms. This imbalance becomes be particularly evident as more annotators are involved. We cannot just include every revision of the annotated sets, since that would mean that the shared dataset is weighed higher than the rest. Conversely we cannot generate a single set representative of an 'average' annotator, because only entries that are unanimously classified as matching one of the two classification categories can be used.

Annotator Agreement

Since the reading speed and annotation process varied from one annotator to another, not all of them produced outputs of equivalent size. The annotators did, however, proceed through the instances of epanaphora in the input dataset in a sequential order, therefore maximising the number of instances of epanaphora that were handled by all annotators.

Of the shared set of epanaphora, a total of 156 unique instances of epanaphora were annotated by all annotators. Out of those 156, only two were marked by all annotators as being true intentional epanaphora. A further 132 were marked by none of the annotators as belonging to the intentional epanaphora output set, and the remaining 22 were marked with varying degrees as belonging to either the intentional epanaphora or the accidental epanaphora groups. Based on these numbers, the annotators' results agree across the board 85.897 percent of the time. More importantly, only 1.282 percent of the recorded instances of epanaphora are unanimously marked are being intentional, and 15.385 percent of the instances of recorded repetitions are considered by at least one person as being intentional. All these numbers are shown in Figure 5.4.

Of all the values in Figure 5.4, three are of higher interest to us than the rest: The 85.897 percent of unanimously annotated epanaphora, the 15.385 percent of intentional epanaphora, and the 1.282 percent of unanimously annotated intentional epanaphora. The first value gives us a rough idea for the base value to aim towards

Description	Instances	Percent
Intentional (agreed)	2	1.282
Unintentional (agreed)	132	84.615
Intentional (debated)	22	14.103
Agreed	134	85.897
Intentional (any)	24	15.385
Total	156	100

Figure 5.4: Statistics on inter-annotator agreement

when performing automated epanaphora classification. The second value gives us an estimate for the ratio of intentional to accidental epanaphora in the corpus, and the third value shows just how sparse the amount of unambiguously intentional epanaphora can be.

Annotation Attribute Analysis

Having looked at annotator agreement, we can now focus our attention on analysing the attributes of the instances of epanaphora used to train the classifiers. Our primary goal in this section is to look at the distribution values for each attribute.

The first attribute we shall examine is n-grams. Figure 5.5 shows the comparative distribution of annotated instances of epanaphora that match the criteria for intentional epanaphora and those that do not match those criteria. The values represent the median n-gram length for each annotated paragraph, and have been normalised to be independently representative of the distribution for the matching and non-matching sets, respectively. In other words the given values are comparative to the distribution for an equal-size sample of each of the annotation sets. The first thing that can be observed is that the distribution of entries which do not match the cri-

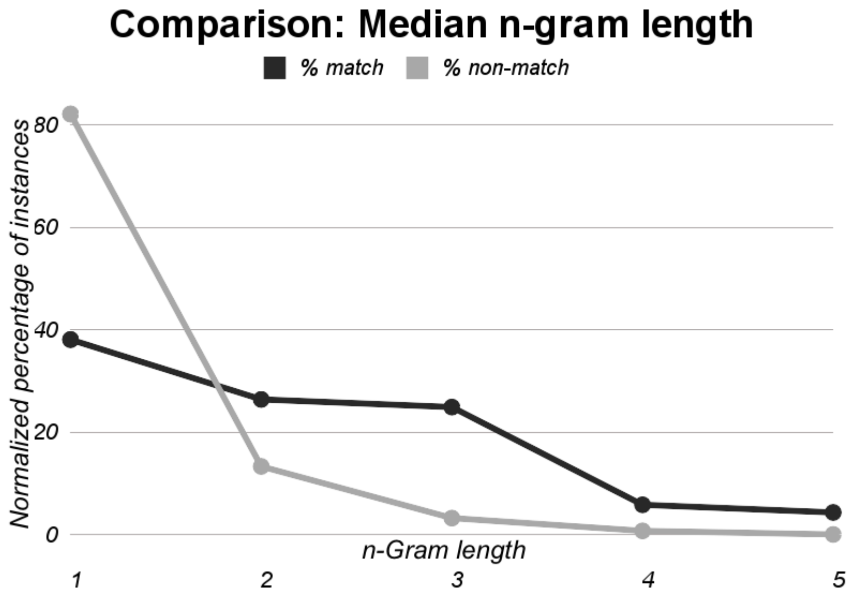


Figure 5.5: n-Gram statistics

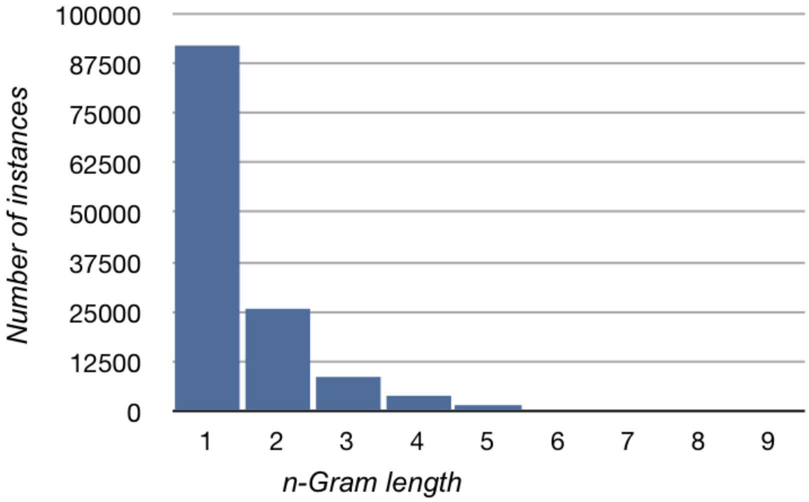


Figure 5.6: Distribution of epanaphora in input corpus by n-gram length

teria for intentional epanaphora mirror the distribution of epanaphora in the input corpus in Figure 5.6. The values from this figure were previously used to obtain the (logarithmic) distribution for Figure 3.9 in Section 3.2.4, and are related in the same fashion as the values for Figure 3.6 and Figure 3.7 in Section 3.2.3.

More interesting than the distribution of accidental epanaphora is the distribution of intentional epanaphora. Figure 5.5 shows a clear bulge in the distribution for 3-grams. The protuberance coincides with the trends described in Section 3.2.3 where it was discovered that instances intentional epanaphora tend to favour the lower end of the n-gram spectrum. We can therefore state with good confidence that the predictions laid out then have been corroborated in Figure 5.5. A second result from the distribution of intentional epanaphora in Figure 5.5 is the y-values where the n-gram length η equals to one. For the described case our distribution graph the relative value of intentional epanaphora is higher than for longer n-grams. However, despite being the largest value for that particular set it is evident that the instances of epanaphora with an n-gram length of one are overwhelmingly accidental.

The second primary attribute described in Chapter 3 is the tuple width. Figure 5.7 shows the comparative distributions for tuple widths in the annotated sets of epanaphora. As was the case with n-gram length, the tuple width for accidental epanaphora mirrors the distribution of the input corpus, seen in Figure 5.8. Furthermore the distribution of intentional epanaphora in Figure 5.7 rather faithfully mirrors that of accidental epanaphora, save for a prominent spike for epanaphora of width four. Most interesting though is the fact that the curve for the distribution of intentional epanaphora goes flat for tuples of width six and larger, as opposed to the

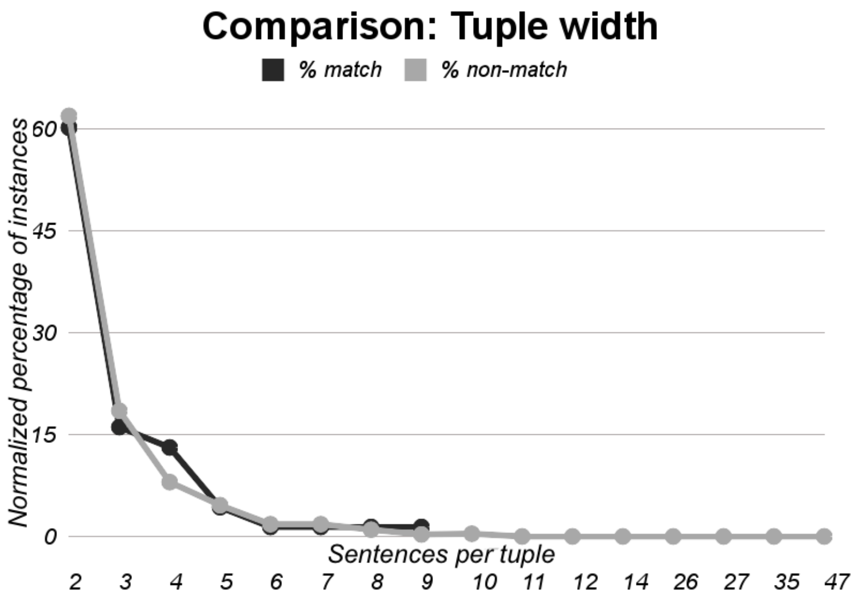


Figure 5.7: Tuple statistics

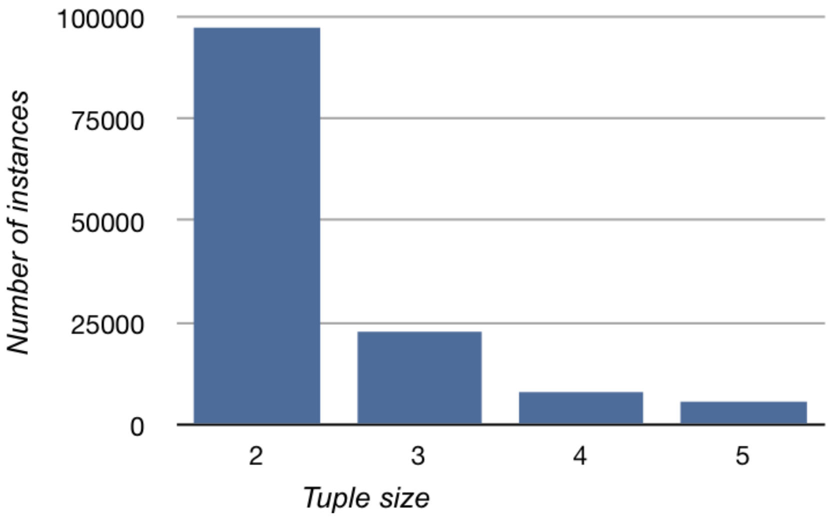


Figure 5.8: Distribution of epanaphora in input corpus by tuple width

distribution curve for accidental repetitions which continues to decay. Unfortunately our data at this stage of our research is too sparse to determine whether the flattening is a general trend or just an anomaly for this particular dataset. While it is tempting to discard it as an anomaly we have encountered clear counter-examples. A possible justification for categorising it as a trend is that as repetitions become wider they also cover a larger proportion of their respective paragraphs. While narrow repetitions are easy to shrug off as random, wider repetitions become more noticeable to the author and audience, and as a result they also become more likely to be intentional. The increased likelihood of intentionality as the width increases may be sufficient to counter the natural logarithmic decay in frequency.

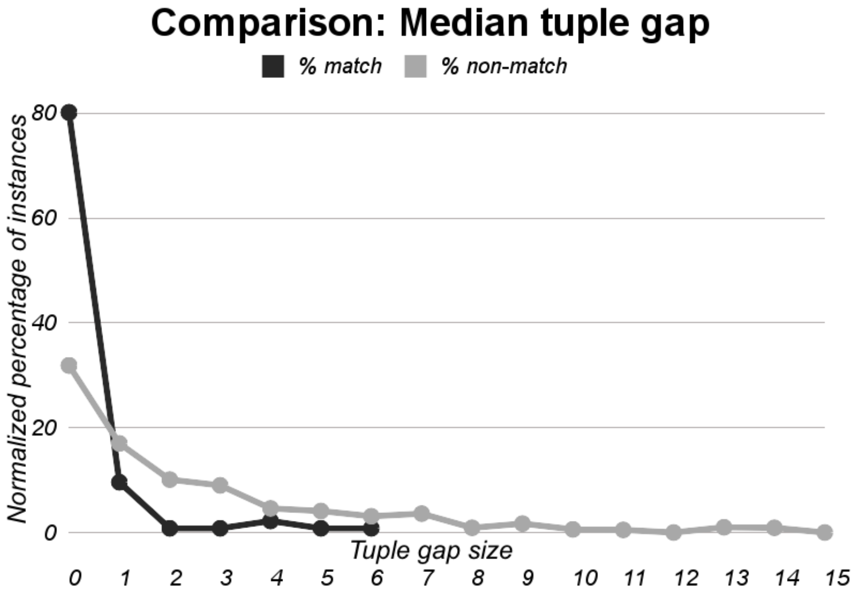


Figure 5.9: Gap statistics

Having examined the two primary classification attributes – n-gram length and tuple width – we can now shift our focus to the secondary attributes. We have just concluded covering tuple width, therefore it is reasonable to continue with a related attribute, namely the gap width between tuples. Figure 5.9 shows the comparative distributions of gaps for intentional and accidental epanaphora respectively. Unfortunately we do not have a baseline value to compare against. However, based on the trends for n-gram length and tuple width we can assume with reasonable safety that the distribution of gaps for accidental epanaphora closely mirrors that of the input corpus.

The most prominent revelation from Figure 5.9 is the abrupt decline in frequency of intentional epanaphora as the size of the gaps increases. Based on the trend observed in this graph, gaps of size two or larger are very strong indicators of the repetition being accidental.

Keywords are the other secondary attribute to be examined. Unlike the previous analyses of attribute distribution, keywords have no specific relation to one another. It is thus that we will switch our distribution graph for this attribute from a line graph to a bar graph, as shown in Figure 5.10. The bars are to be read pairwise, with each pair corresponding to a different keyword. Of all the encountered keyword pairs, four stand out in that they show a marked cleft in the ratio of intentional to accidental epanaphora. These pairs are *articles*, *self identity*, *question keywords*, and *other*. The cleft for the *articles* keyword category is unsurprising, since we already predicted in Section 4.2.4 that repetitions beginning with articles would be largely accidental in nature, in particular if they are paired with an n-gram attribute of short length. What was surprising, however, is the low ratio of keyword repetitions

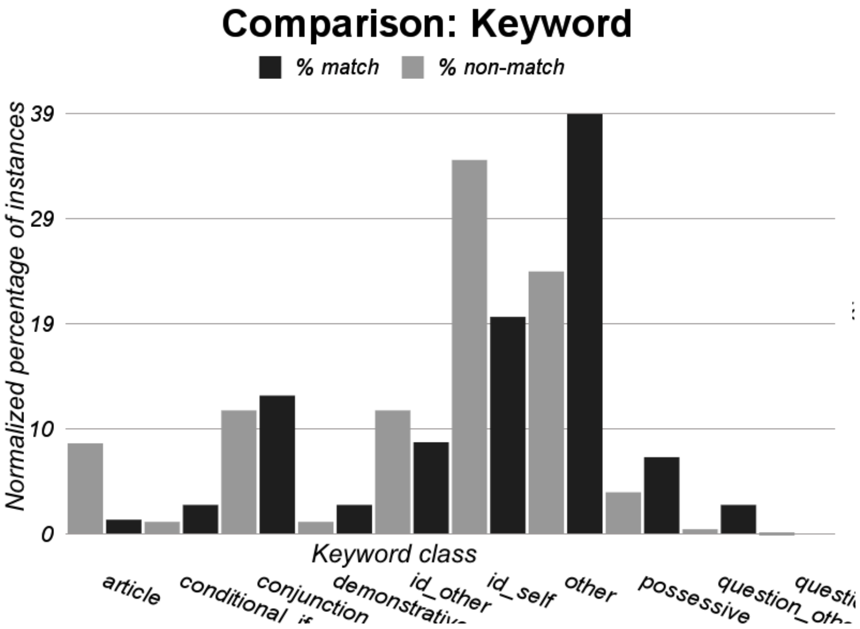


Figure 5.10: Keyword statistics

within the annotated corpus. This low ratio is encouraging, however, as it indicates that this category is less likely to produce a grand number of accidental epanaphora to sift through.

The results for the *self identity* keyword category is quite interesting. As predicted in Section 4.2.1 the first person singular personal pronoun occurs with very high frequency within the input corpus. We did not, however, expect the ratio of accidental to intentional epanaphora to favour the accidental side of the balance. Identity of self is a very strong vehicle for the expression of feelings and desires, which were expected to be one of the larger foundations for intentional epanaphora. The results seen in Figure 5.10 lay a degree of uncertainty over that assumption. One potential explanation is that the input corpus sports an imbalance in the use of self identity. It is possible that a very high incidence of recounts of personal events is responsible for tipping the scales towards the side of accidental epanaphora.

The third keyword category, *question keywords*, promises a brilliant insight into the use of repeating questions. Despite being a relatively small category in itself, when comparing the incidence of question keywords related to intentional epanaphora to that of question keywords related to accidental epanaphora it is clear that intentional epanaphora are favoured prominently. This favouritism is very likely due to the strong use of repetition in rhetorical questions. Further speculation is not possible due to the sparsity of the input corpus, but if Figure 5.10 is any indicator of future trends then question keywords are certainly something to look out for in future iterations of epanaphora detection and classification.

The last keyword category, *other*, is not so much an intentional category as a container for instances of epanaphora that do not match any of the pre-set categories of keywords. What makes it interesting is that the ratio of intentional epanaphora marked as ‘other’ is significantly large compared to the same ratio for accidental epanaphora. Two explanations are possible for the prominence of this ‘other’ cluster. The simplest answer is that the other defined categories are insufficient for proper keyword categorisation, and that further sub-sectioning is required. The other explanation is that the defined categories are very good at identifying accidental epanaphora and act as a filter to discard unwanted repetitions.

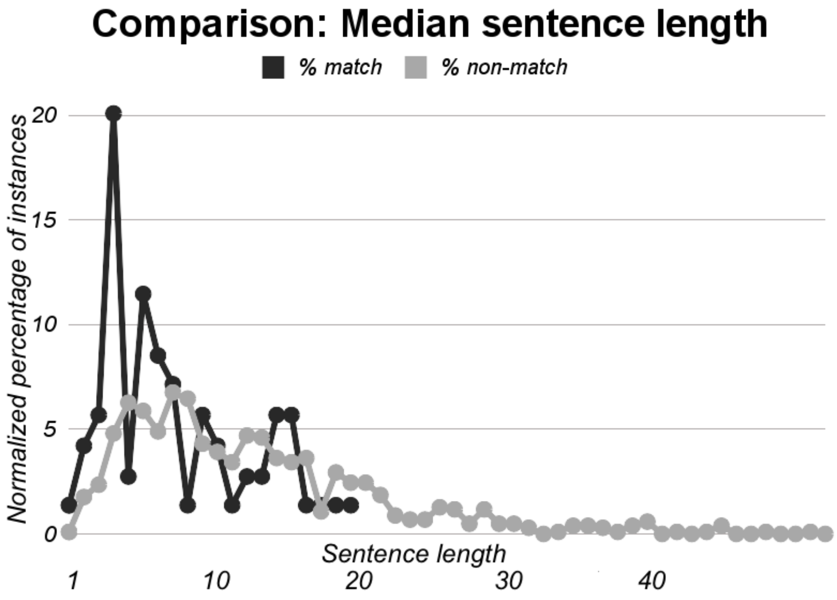


Figure 5.11: Epanaphora sentence length statistics

Despite having already defined and set the primary and secondary attributes, during the process of evaluating the annotated corpus we got interested in one additional variation, namely the length of the sentences (in words per sentence) that were involved in intentional epanaphora. Figure 5.11 shows the plot of the values for this attribute, and it shows a very interesting trend. Accidental epanaphora (and, in turn, the input corpus) show a small but quick spike in frequency of repetitions as the sentence length increases, followed by a slow and steady decline for even longer sentences. On the other hand the data plot for intentional epanaphora displays a large spike early on, followed by a very prominent decay. The shape of the curve for intentional epanaphora in Figure 5.11 hints that it is crucial for the repetitions in intentional epanaphora to follow each other in quick succession within the source text. The results from Figure 5.11, together with those from Figure 5.9, can be seen as a corroboration for the idea in Section 4.1.1 that *memory span* plays an important role in the perceived intentional nature of epanaphora. Given the apparent importance of memory span, its correlation with the perceived intentional nature of epanaphora bears formalising:

Hypothesis 5.1.1. *Memory span has a key influence in the perceived intentional nature of epanaphora.*

If memory span can be used to judge the intentional nature of epanaphora, then we surmise that it is likely that this relation also works the other way. We can further extend that conjecture by tying back it into one of the premises of this thesis:

Hypothesis 5.1.2. *The detection and classification of intentional epanaphora be used to measure the saliency of a section of text.*

5.1.2 Classifiers

The tools that was used to train and run the epanaphora classifiers is Weka [76][63][148]. Weka is a Java-based classification tool built at the University of Waikato and released under the GNU General Public License. It provides a common interface to a large variety of machine learning and data mining algorithms. In addition to allowing users to train and test their classifiers, Weka is also capable of generating vital graphs and statistics necessary to evaluate those classifiers.

Classifier Types

Before proceeding to the training of the epanaphora classifiers it is advised to look at the kinds of classifiers available and determine which ones are most suitable to the task at hand. It is at this point that the additional effort from Section 4.3 pays off. In said section we had studied the properties of the various attributes which we intended to use to detect and classify epanaphora. By learning how these attributes interact with one another we can make an educated guess as to which kind of classification algorithm would be most appropriate.

In order to maintain input and output consistency we shall focus our study of classification algorithms on those which are available within Weka. For the first classification effort we are not so much interested in getting the best results as we are in testing a wide variety of classification methods. As a result we will be examining the following algorithms in this section:

- Decision trees

- Support vector machines (SVMs)
- Logistic regression
- Bayesian classification
- Adaptive boosting (ADABoost)

In the decision tree learning method the approach to classification is to create one node for each of the classification attributes and to form a branch from that node for each of the possible values of said attribute. This method allows researchers to perform a comprehensive search through all the variations of the input dataset. Such an approach is useful for sets of relatively small variation. This, however, is not the case in our study. In Section 5.1.1 we showed that each attribute can significantly vary in value. Furthermore we do not have strict upper bounds for any of the numeric attributes. While this is not a terminal problem at run-time, it does make it more difficult to predict the classification outcome for an unknown input set.

Support vector machines are a method of pattern recognition. The premise is that this algorithm is capable of sorting input into one of two categories. This is achieved by mapping each element of the training input as a point in space in such a manner that a ‘hyperplane’ (or set thereof) can be drawn through said space to separate the points belonging to each individual category. The best classification is achieved by finding the set of hyperplanes which maximise the separation between the two output categories, that is, the hyperplanes which aim for the maximum distance to the nearest datapoints. For the purpose of our study the binary classification is ideal, since it matches the method used by the annotators during their phase of the classification procedure. However, it is not clear whether the list of attributes for classification is appropriate for classification by support vector machines. Given that intentional epanaphora can cover a variety of value ranges for each attribute it is possible that the datapoints do not distribute well in the classification space, and that as a result there is no hyperplane with sufficiently large distance to the nearest datapoints to compensate for generalisation errors.

Logistic regression is a probabilistic model for classification. Prediction can be performed on numeric values as well as literals, and is achieved by plotting the data to a logit function logistic curve. Because of that it is a good fit for our research, since the function can map input values z of any magnitude to an output $f(z)$ value between zero and one. The largest disadvantage in utilising logistic regression as an epanaphora classifier is that with small or medium training sets the model has a tendency to overestimate effect ratios. This overestimation falls within the acceptable error ranges for a single execution, but becomes more prominent in multi-part studies.

As with logistic regression, Bayesian classifiers are probability-based. The major change compared to the previous methods – decision trees, support vector machines, and logistic regression – is that in classification using Bayesian methods each attribute is considered to contribute independently to the probability of the classification. In this pilot study we focused specifically on using a naive Bayes classifier. This type of model is well-suited to supervised learning, even when dealing with relatively small training corpora. This gives it an advantage over the other probabilistic classifier used, logistic regression. Classification is also performed based on maximum likelihood on a per-variable basis, making the algorithm simple and robust since it is not necessary to determine the entire covariance matrix. Epanaphora classification using a naive Bayes algorithm is well-suited to the setup of our study since we have explicitly recorded any direct relations between the attributes used for the classification of instances of epanaphora. Furthermore the use of a naive Bayes classifier addresses the issue previously mentioned during the discussion of support vector machine classification, namely the distribution of intentional instances of epanaphora across a large range of attribute values. By performing classification as the sum of independent probabilities we side-step that issue entirely.

Classifier Training Data

As mentioned in Section 5.1.1, the annotated data has been split into two sets: One for measuring annotator agreement, and one for training of the classification algorithms. However a training dataset alone is not sufficient. After training, the accuracy of the classification algorithm needs to be measured. There are two possible approaches to generating a test set for the classification algorithms. The first option is to split the training dataset in two sections - One for training, and another one for testing. The second option is to use the dataset to measure annotator agreement. The latter seems counter-intuitive, but the idea of using this dataset is actually not that far-fetched. Each epanaphora classification algorithm can be construed as being an additional annotator, and in that light it is reasonable to be using the given dataset for testing.

Still, one more aspect of the training corpus has to be addressed, and that is the uneven ratio of intentional to accidental epanaphora in the training corpus. In

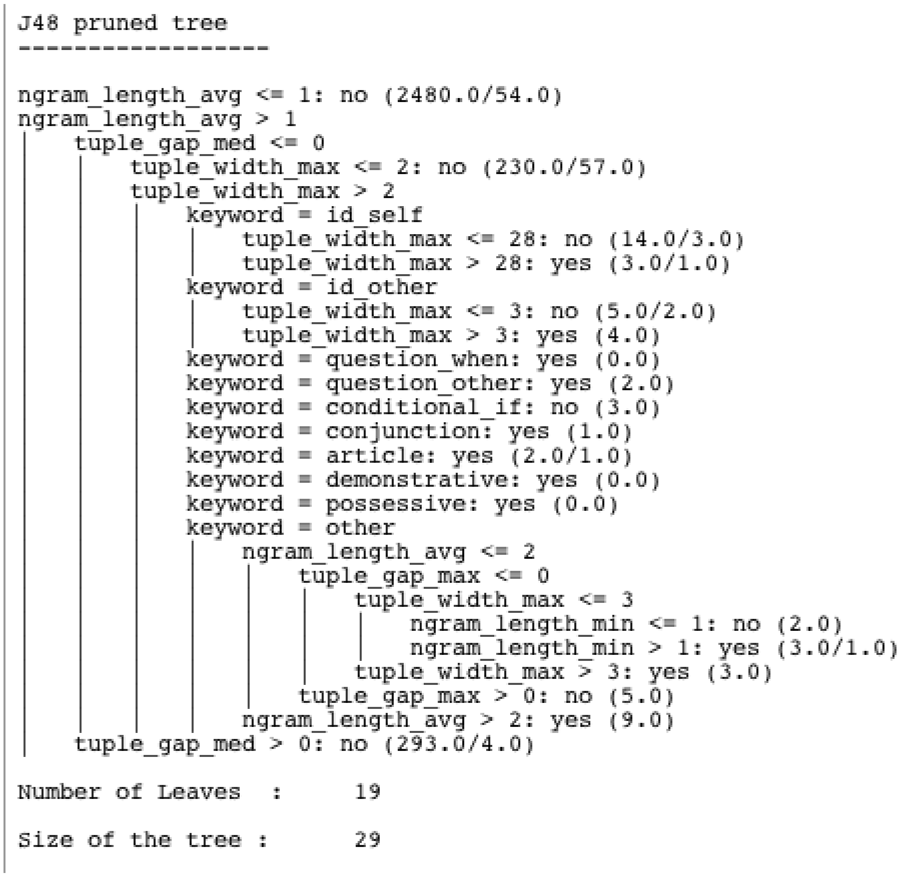
Section 5.1.1 we discovered that the number of accidental epanaphora is between one or two orders of magnitude greater than that of intentional epanaphora. Depending on the nature of the data and the classification schemas this may not be an issue. However there is a possibility that the given imbalance in numbers may bias the classification algorithms towards tagging a higher number of epanaphora as accidental than they would otherwise. This asymmetry can have a particularly strong effect on non-stochastic classification algorithms which are more affected by overlapping values for intentional and accidental epanaphora. In order to address this possible condition we will run the classifier training with two separate sets of input – One with the unadulterated training data, and and one input in which the training corpus is balanced in such a way that there is an equal number intentional and accidental instances of epanaphora.

5.1.3 Pilot Test

Having finalised the selection of the classifier algorithms and chosen the datasets for training and testing said algorithms we can move forward with the training procedure. We will first study the classification results based on the original annotated training set. Each result table consists of the following ratios: True Positive (TP), False Positive (FP), Precision, Recall, F-measure, and Receiver Operating Characteristic (ROC) area values for each individual classification class and for the weighed average.

In order to maintain consistency we will proceed through the classification schemas in the order listed in Section 5.1.2. For the decision tree classifier we chose the J48 algorithm, and for the support vector machine classifier we used the libSVM library.

Training Results



```
J48 pruned tree
------------------

ngram_length_avg <= 1: no (2480.0/54.0)
ngram_length_avg > 1
|   tuple_gap_med <= 0
|   |   tuple_width_max <= 2: no (230.0/57.0)
|   |   tuple_width_max > 2
|   |   |   keyword = id_self
|   |   |   |   tuple_width_max <= 28: no (14.0/3.0)
|   |   |   |   tuple_width_max > 28: yes (3.0/1.0)
|   |   |   keyword = id_other
|   |   |   |   tuple_width_max <= 3: no (5.0/2.0)
|   |   |   |   tuple_width_max > 3: yes (4.0)
|   |   |   keyword = question_when: yes (0.0)
|   |   |   keyword = question_other: yes (2.0)
|   |   |   keyword = conditional_if: no (3.0)
|   |   |   keyword = conjunction: yes (1.0)
|   |   |   keyword = article: yes (2.0/1.0)
|   |   |   keyword = demonstrative: yes (0.0)
|   |   |   keyword = possessive: yes (0.0)
|   |   |   keyword = other
|   |   |   |   ngram_length_avg <= 2
|   |   |   |   |   tuple_gap_max <= 0
|   |   |   |   |   |   tuple_width_max <= 3
|   |   |   |   |   |   |   ngram_length_min <= 1: no (2.0)
|   |   |   |   |   |   |   ngram_length_min > 1: yes (3.0/1.0)
|   |   |   |   |   |   tuple_width_max > 3: yes (3.0)
|   |   |   |   |   tuple_gap_max > 0: no (5.0)
|   |   |   |   ngram_length_avg > 2: yes (9.0)
|   tuple_gap_med > 0: no (293.0/4.0)

Number of Leaves  :     19

Size of the tree :      29
```

Figure 5.12: J48 decision tree

We will first examine the results for the classifiers trained with the unedited input corpus. Figure 5.12 shows the decision tree that was built from the training corpus. As we proceed down the tree we can see right away that this training corpus is not compatible with the algorithm. The classification values for average n-gram length, median tuple gap, and tuple width are ignored by the algorithm. The reason for this is that these attributes have no clear split value. Plotting the ROC curve (Figure 5.14) shows a graph that has nearly no predictive value. The effect of this can be seen on the accuracy table on Figure 5.13. The first row shows the values for epanaphora classified as intentional, the second row for epanaphora classified as accidental, and the third row shows the weighed average. We can see from the TP and FP rates that not a single instance of epanaphora has been rated as intentional. This is not well-reflected in the precision and recall measures due to inherent imbalance in the ratio of intentional to accidental epanaphora in the corpus, but can be seen clearly in the low value of the retriever operating characteristic (ROC) area.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0	0	0	0	0	0.606	yes
1	1	0.886	1	0.94	0.606	no
0.886	0.886	0.785	0.886	0.833	0.606	avg

Figure 5.13: J48 decision tree accuracy by class

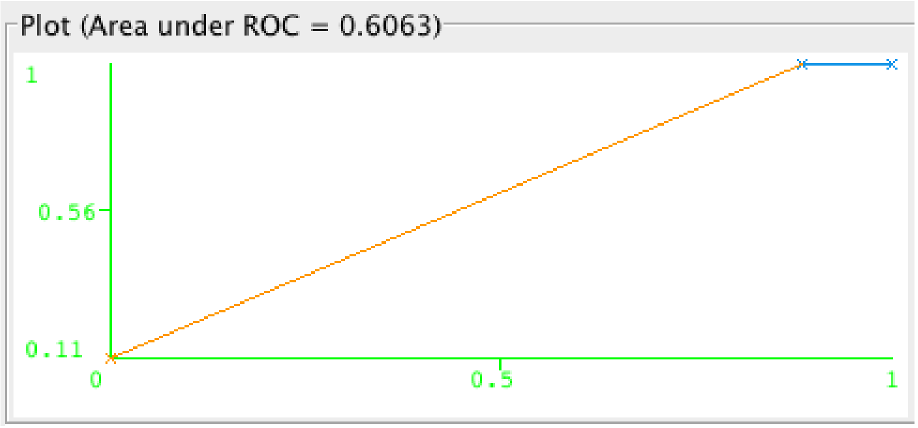


Figure 5.14: Receiver operating characteristic (ROC) curve for the J48 decision tree

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0	0	0	0	0	0.5	yes
1	1	0.886	1	0.94	0.5	no
0.886	0.886	0.785	0.886	0.833	0.5	avg

Figure 5.15: LibSVM support vector machine accuracy by class

Classification via support vector machine did not perform any better. In fact this classifier was completely unable to perform any significant prediction, as can be seen from Figure 5.15.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0	0	0	0	0	0.734	yes
1	1	0.886	1	0.94	0.734	no
0.886	0.886	0.785	0.886	0.833	0.734	avg

Figure 5.16: Logistic regression accuracy by class

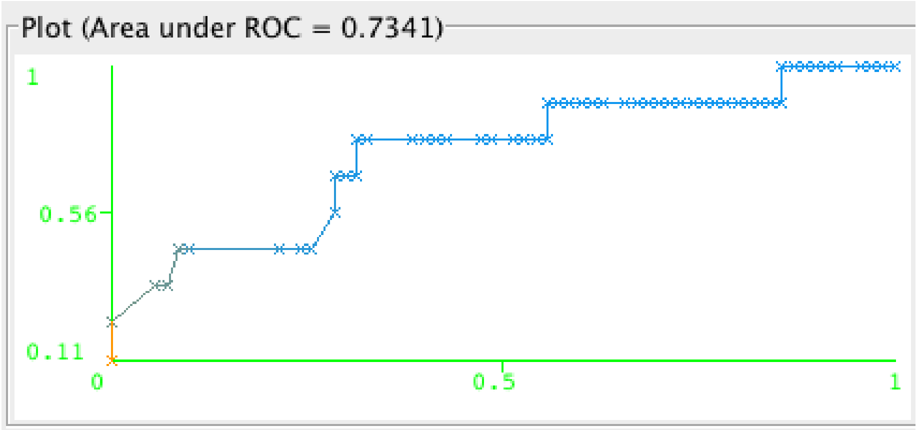


Figure 5.17: Receiver operating characteristic (ROC) curve for logistic regression

Figure 5.16 shows the accuracy by class for the logistic regression classifier. While the true positive and false positive values for logistic regression do not seem to indicate that this classification algorithm will perform any better than its predecessors, the receiver operating characteristic tells a different story. In Figure 5.17 we can see that despite having a zero positive result rate the curve is beginning to show an inclination towards better classification.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.111	0.014	0.5	0.111	0.182	0.759	yes
0.986	0.889	0.896	0.986	0.939	0.752	no
0.886	0.789	0.851	0.886	0.853	0.753	avg

Figure 5.18: Naive Bayes accuracy by class

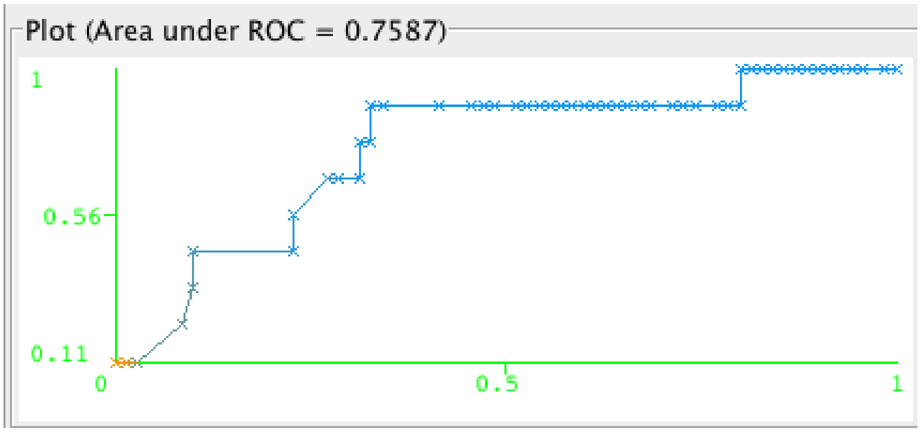


Figure 5.19: Receiver operating characteristic (ROC) curve for Bayesian classification

Of all the classification algorithms trained during the pilot, the naive Bayesian classifier was the only one to categorise some instances of epanaphora as intentional (Figure 5.18). This is a major breakthrough in that it promises that the naive Bayes classification algorithm is capable of more fine-grained distinction between intentional and accidental epanaphora using the given attributes. This is backed by the ROC curve (Figure 5.19), which shows greater convexity than the previous best one for logistic regression.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.111	0	1	0.111	0.2	0.763	yes
1	0.899	0.897	1	0.946	0.763	no
0.889	0.788	0.909	0.899	0.861	0.763	avg

Figure 5.20: Naive Bayes accuracy by class (including the sentence length attribute)

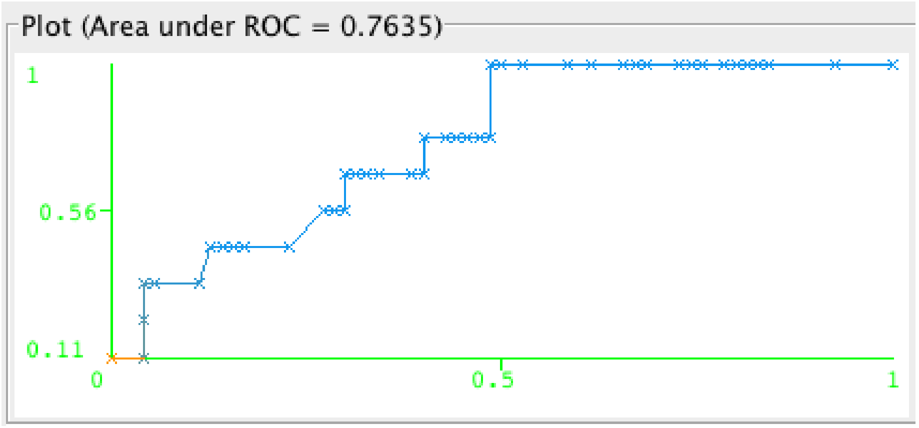


Figure 5.21: Receiver operating characteristic (ROC) curve for Bayesian classification (including the sentence length attribute)

In Section 5.1.1 we had observed the apparent effect of the sentence length on the intentionality of epanaphora. This attribute of the annotated epanaphora has not been included in the results studied until now. However, given the possibility that it may have a positive effect on the quality of epanaphora classification we felt that it is necessary to introduce an additional round in this study to examine the effect that this attribute may have on the classification algorithms. Doing so showed no change for the decision tree, support vector machine, and logistic regression classifiers in their ability to detect intentional instances of epanaphora. The naive Bayes classifier, however, performed again with better results than the rest of the algorithms. The results of that classifier execution can be seen in Table 5.20. We can see that the algorithm was able to maintain the true positive rate for intentional epanaphora while eliminating the false positive rate for the same class. This translates into a perfect precision measure while maintaining the recall rate, which improves the resulting f-measure. Finally the receiver operating characteristic area sees further gain. The ROC curve shows greater concavity and a smoother curve (Figure 5.21). All of the above numbers appear to support Hypothesis 5.1.2 in that a it appears to show the existence of a certain range for the sentence length which sports an increased incidence in intentional epanaphora. However, since the naive Bayes classifier is the one most sensitive to the given attributes it is the only one that has been able to take advantage of the new sentence length information.

Having evaluated the success of training the classification algorithms with the unadulterated training corpus we will now proceed to using the balanced training dataset. The balance is achieved by randomly eliminating entries from the larger of the two annotation categories (accidental epanaphora) until the number of instances of epanaphora encountered in it equals to the lesser of the annotation categories, intentional epanaphora. This significantly reduces the size of the training corpus. Depending on the frequency of annotation of instances of epanaphora as intentional the size of the corpus is reduced by one or two orders of magnitude. As already observed in Section 5.1.2, some of the classification algorithms are more error-prone with smaller training corpora than others. Finally, since we have determined that the length of the constituent sentences of an instance of epanaphora have a mostly positive influence on the classification of unknown epanaphora we have decided to include those values in this classifier training round.

The decision tree classifier is finally capable of generating a distinction between the intentional and accidental instances of epanaphora based on the provided attributes. The tree constructed by the J48 algorithm can be seen in Figure 5.24. From the view we can observe that the input corpus sports a much clearer separation between the values for intentional and accidental epanaphora, as evidenced by the branching for maximum n-gram length, average tuple gap, and median and average sentence lengths. Surprisingly the tuple width did not make it into the pruned decision tree. We speculate that this is because this attribute of epanaphora is not singly characteristic for either intentional nor accidental epanaphora. Figure 5.22 shows a clear improvement for the decision tree classifier with the balanced training corpus. However it is evident that this balanced training corpus is not sufficient to place the decision tree algorithms above even the naive Bayes classifier trained with the unadulterated corpus with the sentence length attribute. While the decision tree

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.222	0.1	0.222	0.222	0.222	0.682	yes
0.9	0.778	0.9	0.9	0.9	0.682	no
0.823	0.701	0.823	0.823	0.823	0.682	avg

Figure 5.22: Decision tree accuracy by class (balanced corpus, including the sentence length attribute)

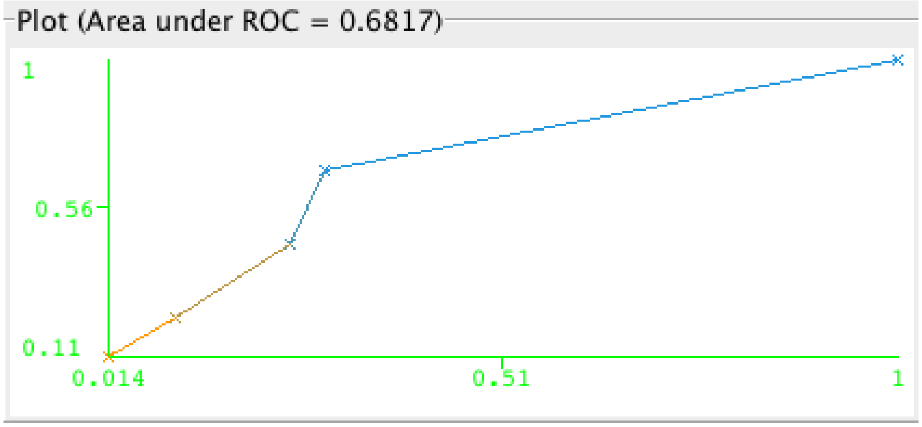
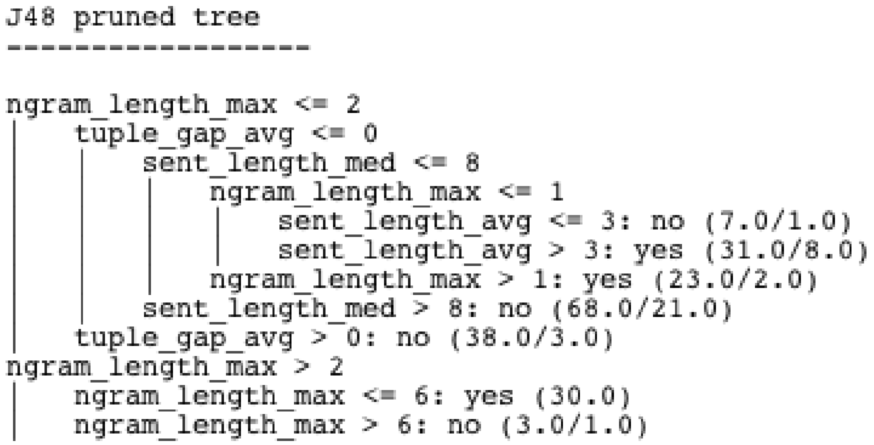


Figure 5.23: Receiver operating characteristic (ROC) curve for decision trees (balanced corpus, including the sentence length attribute)



```
J48 pruned tree
------------------

ngram_length_max <= 2
|   tuple_gap_avg <= 0
|   |   sent_length_med <= 8
|   |   |   ngram_length_max <= 1
|   |   |   |   sent_length_avg <= 3: no (7.0/1.0)
|   |   |   |   sent_length_avg > 3: yes (31.0/8.0)
|   |   |   ngram_length_max > 1: yes (23.0/2.0)
|   |   sent_length_med > 8: no (68.0/21.0)
|   tuple_gap_avg > 0: no (38.0/3.0)
ngram_length_max > 2
|   ngram_length_max <= 6: yes (30.0)
|   ngram_length_max > 6: no (3.0/1.0)
```

Figure 5.24: Pruned J48 decision tree (balanced corpus, including the sentence length attribute)

classifier is finally capable of annotating intentional epanaphora and doing so with relative success – as shown by the larger true positive rate and f-measure – the area below the ROC plot is still rather small. Looking at the curve itself shows why: In Figure 5.23 we see a slightly convex curve that is still struggling to move away from the diagonal.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.222	0.186	0.133	0.222	0.167	0.518	yes
0.814	0.778	0.891	0.814	0.851	0.518	no
0.747	0.71	0.804	0.747	0.773	0.518	avg

Figure 5.25: Support vector machine accuracy by class (balanced corpus, including the sentence length attribute)

While the decision tree algorithm has shown some visible improvement, the support vector machine classifier has remained the worst under-performer. A higher false positive rate, lower precision and f-measure in Figure 5.25, and a very shallow ROC curve that barely deviates from the diagonal are yet more indicators that support vector machines may not be appropriate for classification of intentional and accidental epanaphora.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.444	0.186	0.235	0.444	0.308	0.648	yes
0.814	0.556	0.919	0.814	0.864	0.648	no
0.772	0.513	0.841	0.772	0.8	0.648	avg

Figure 5.26: Logistic regression accuracy by class (balanced corpus, including the sentence length attribute)

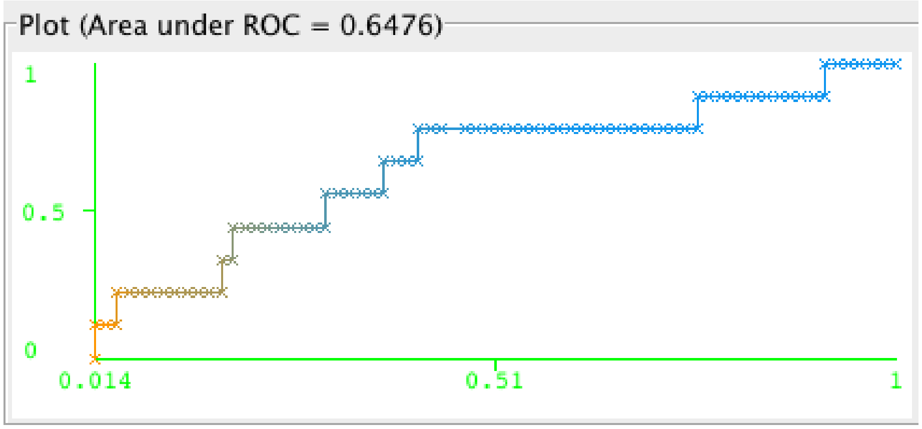


Figure 5.27: Receiver operating characteristic (ROC) curve for logistic regression classification (balanced corpus, including the sentence length attribute)

Like the J48 decision tree classifier, the logistic regression algorithm shows visible improvement. In fact Figure 5.26 indicates that with this algorithm we have the greatest true positive rate yet and a false positive rate that falls within the acceptable error range. Precision is still relatively low, but high recall (and as a result, f-measure) promise to make this algorithm a good candidate for epanaphora classification. However, the ROC plot in Figure 5.27 tells a different story, with a curvature that is less convex (though better-staged) than that of Figure 5.17.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.333	0.143	0.231	0.333	0.273	0.798	yes
0.857	0.667	0.909	0.857	0.882	0.798	no
0.797	0.607	0.832	0.797	0.813	0.798	avg

Figure 5.28: Naive Bayes accuracy by class (balanced corpus, including the sentence length attribute)

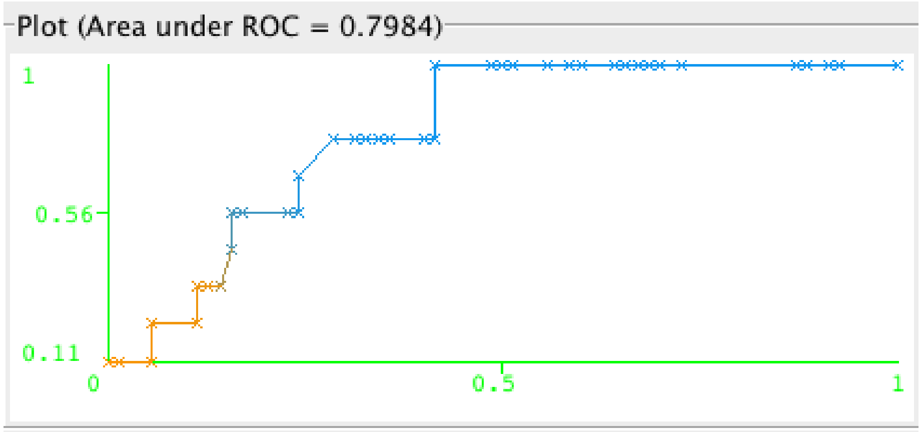


Figure 5.29: Receiver operating characteristic (ROC) curve for Bayesian classification (balanced corpus, including the sentence length attribute)

Finally we have the results from training the naive Bayes classifier with the balanced corpus including sentence lengths in Figure 5.28. The true positive rate sees a decline in comparison to the logistic regression algorithm, but it is greater than the previous results for the Bayesian classifier and it also sports a lower false positive rate, comparable precision, and the second best f-measure. The real treasure however is the the area under the ROC curve, with the highest ratio of all variations tested so far. Looking at the shape of the curve in Figure 5.29 we can immediately see why – the curve has a steep incline and high convexity.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.778	0.286	0.259	0.778	0.389	0.755	yes
0.714	0.222	0.962	0.714	0.82	0.75	no
0.722	0.229	0.882	0.722	0.771	0.751	avg

Figure 5.30: Naive Bayes accuracy by class (balanced corpus, excluding the sentence length attribute)

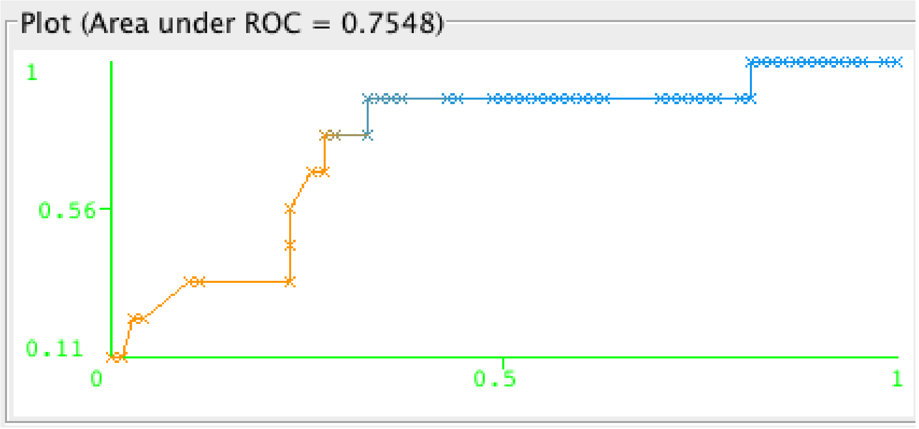


Figure 5.31: Receiver operating characteristic (ROC) curve for naive Bayes classification (balanced corpus, excluding the sentence length attribute)

Last but not least we decided to verify whether the inclusion of sentence length statistics had a similar effect on the balanced corpus than it had on the unedited corpus. For that we ran an additional round with the naive Bayes classifier by training it on the balanced corpus without the sentence length attributes. The result is the accuracy table in Figure 5.30 and the receiver operating characteristic curve in Figure 5.31. The accuracy table shows a greatly increased true positive rate for intentional epanaphora but at the cost of a greater overall error ratio and lower weighed average f-measure. It is apparent that without the sentence attributes the algorithm is not capable of being as discriminatory. Furthermore the area under the ROC plot is smaller than that of the naive Bayes classifier trained with the corpus which included the sentence length attributes. In observing the curve we see that it has a pronounced indentation, and that it tapers off earlier than its counterpart.

We applied the same procedure to the logistic regression algorithm and obtained the ROC curve in Figure 5.33. At a glance we can see that it is smoother than that in Figure 5.27, which is a good sign. However, the accuracy table in Figure 5.32 shows a lower rate of true positives while maintaining a high false positive rate.

We propose that the figures for the two stochastic algorithms appear to support Hypothesis 5.1.2. Balancing the training corpus was definitively the biggest boost in true positive generation, but it was done at the cost of lower precision. The inclusion of the sentence length attributes provided a balancing measure which allowed the algorithms to act more discriminately towards the test corpora. This backs up the distribution observed in Figure 5.11. Without a sentence length discriminator the distribution of epanaphora tagged as intentional reflected that of the original,

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.556	0.257	0.217	0.556	0.313	0.721	yes
0.743	0.444	0.929	0.743	0.825	0.721	no
0.722	0.423	0.848	0.722	0.767	0.721	avg

Figure 5.32: Logistic regression accuracy by class (balanced corpus, excluding the sentence length attribute)

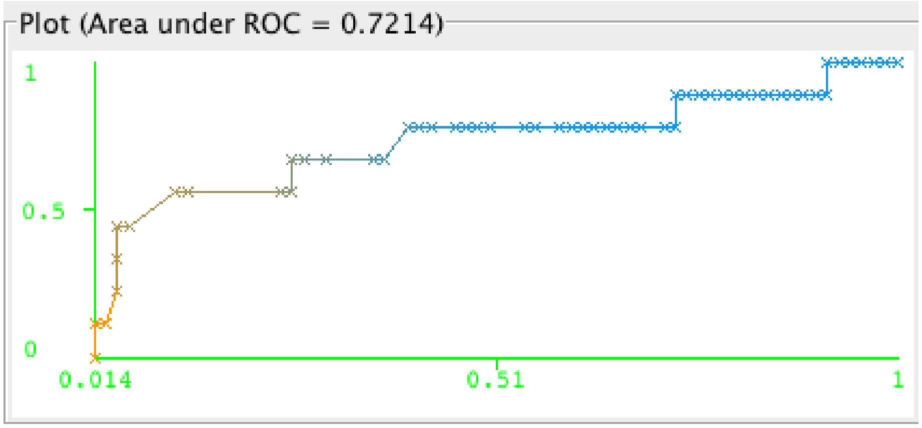


Figure 5.33: Receiver operating characteristic (ROC) curve for logistic regression classification (balanced corpus, excluding the sentence length attribute)

unannotated corpus. By introducing the sentence length attribute we were able to provide a training set that more closely resembled the perceptions of the human annotators.

5.1.4 Constraint Tightening

Having demonstrated that it is possible to train machine learning algorithms to detect likely instances of intentional epanaphora we now want to focus on generating a training set that is geared towards producing classifiers capable of discerning instances of epanaphora which are indubitably intentional. This is a significant focus shift from the previous classification goal. The intent in Section 5.1.2 was to produce a trained classification algorithm with the best results for generic intentional epanaphora. The measurements for that goal were performed on a relatively small corpus, since it required an evaluation step that involved a manually tagged corpus. In this section we move beyond that constraint by working on a large corpus and evaluating a representative sample of the resulting corpus. This allows us to shift our focus towards maximising the precision of the classification algorithm at the cost of recall, since we are no longer intent on maintaining a decent f-measure ratio for intentional epanaphora classification.

Since we are no longer bound by the recall measure on the classification results we are immediately able to tighten the constraints for the generation of the classifier training corpus. The most conspicuous method of achieving this is by pruning those instances of epanaphora which are least likely to be classified as intentional epanaphora. By tagging the instances of epanaphora whose attributes are least likely to lead towards an intentional classification we are able to automatically generate a blacklist during the classifier training process.

Attribute Constraints

In Section 5.1.1 we had already performed an early evaluation of the attributes of intentional and accidental epanaphora, and observed the qualities most likely to produce unambiguous tagging of intentional epanaphora. We can now revise these observations and use them to produce strict pruning rules.

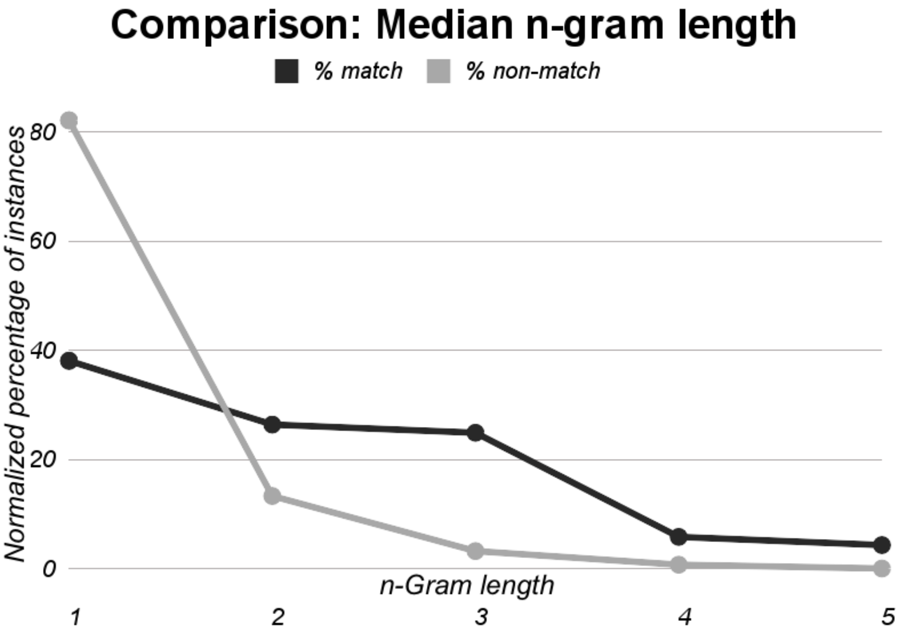


Figure 5.34: Annotated n-gram statistics

The first constraint to be applied is the pruning of instances of epanaphora whose n-gram length is one. An n-gram length of one indicates that only the first word of each sentence involving a particular epanaphora pattern is being repeated. From Figure 5.34 we can clearly see that epanaphora with this n-gram length are overwhelmingly strong representatives of accidental epanaphora.

Observation 5.1.3. *Epanaphora of n-gram length one are least likely to be of intentional nature.*

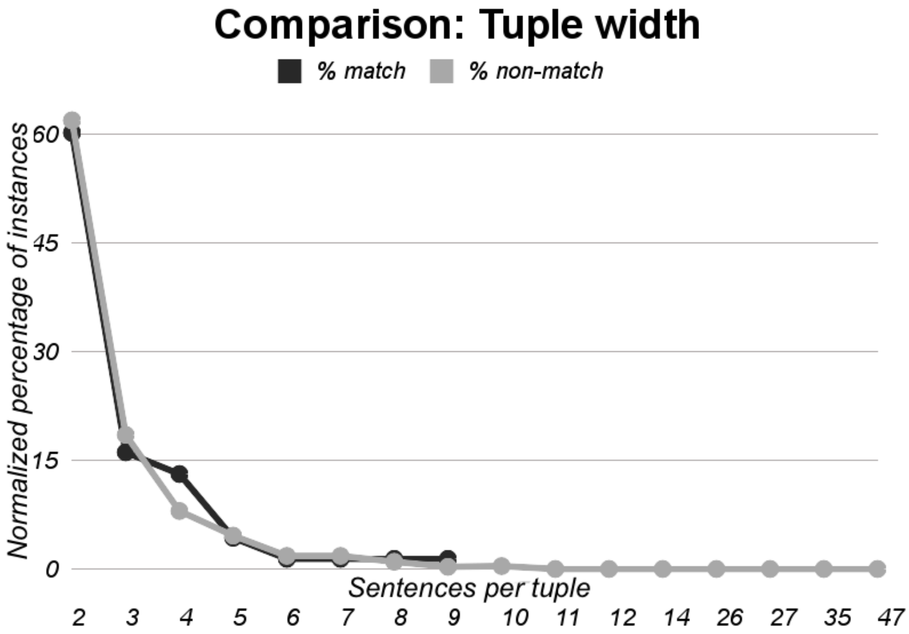


Figure 5.35: Annotation tuple statistics

Unlike n-gram statistics there is no clear indicator in Figure 5.35 that epanaphora with tuples of any width are significantly less likely to be classified as accidental over intentional. Instances of epanaphora of tuple width four are slightly more likely to be classified as intentional, but not overwhelmingly so. Reviewing these statistics helps explain why the J48 decision tree in Figure 5.24 does not have a branch for tuple width. Still, we have to keep in mind that the values shown in Figure 5.35 are normalised over a representative sample of 100 for each category. Referencing back to Figure 5.4 we see that instances of accidental epanaphora are much more pervasive than instances of intentional epanaphora. If we recreate Figure 5.35 with absolute values in place of the normalised statistics we discover that the uneven ratio of intentional to accidental epanaphora generates an imbalance in the incidence of intentional epanaphora as seen in Figure 5.36. In order to amortise this asymmetry we have decided that it is prudent to prune all instances of epanaphora of tuple width two. This step does not alter the normalised ratio of intentional to accidental

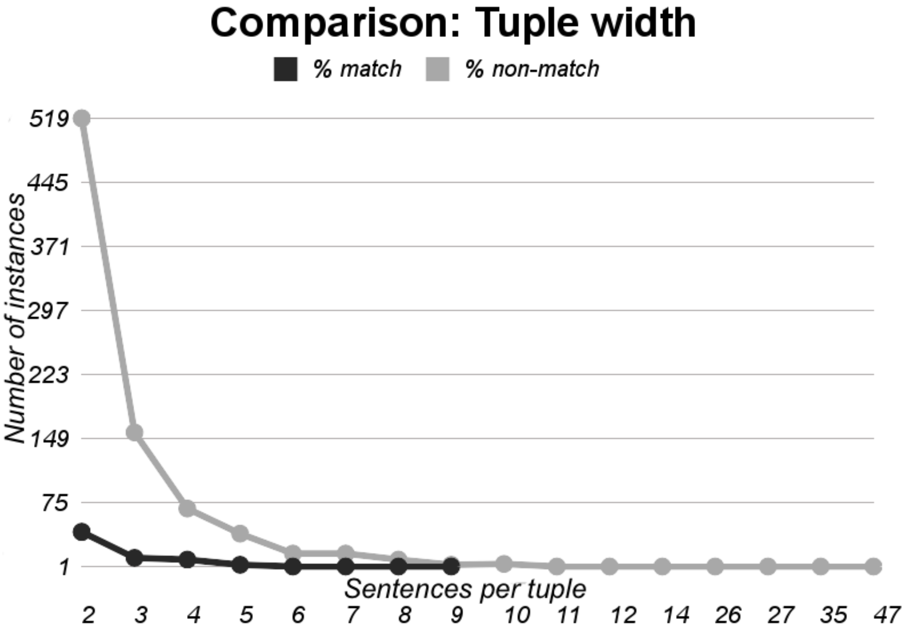


Figure 5.36: Annotation tuple statistics (absolute values)

epanaphora, but it does significantly reduce the number of accidental epanaphora encountered during the annotation process. In doing so we facilitate an increased performance for the routine of annotating instances of intentional epanaphora.

Observation 5.1.4. *There is no uniquely identifying tuple width characteristic which distinguishes between intentional and accidental epanaphora.*

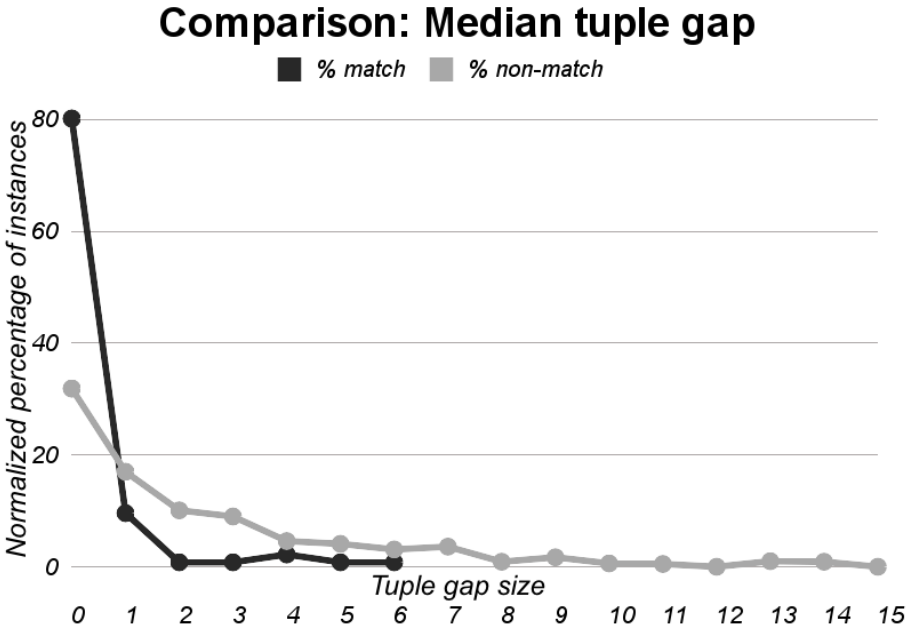


Figure 5.37: Annotation gap statistics

The third constraint based on attribute analysis is for the gap width between constituent sentences of intentional instances of epanaphora. Figure 5.37 shows a very steep decay in the incidence of intentional epanaphora as the median width of the gap increases. It is immediately clear that instances of epanaphora with a median gap width that is greater than or equal to six are accidental beyond the shadow of a doubt. We can furthermore increase this constraint to a reasonable set including instances of epanaphora of median gap width greater than or equal to two. This leaves us with zero-gap epanaphora and instances of epanaphora with a median gap width of one. The former is plainly the best representation of intentional instances of epanaphora. The latter can be judged in two different ways. On the one hand instances of epanaphora of width one are more frequently accidental than intentional. On the other hand they are still a significant portion of all the instances of epanaphora tagged as intentional. Ultimately the deciding factor in deciding whether to include instances of epanaphora matching this attribute value is the initial justification for the criteria pruning process, namely the focus on precision over recall. As a result we have chosen to include epanaphora of a median width of one in the pruning step.

Observation 5.1.5. *A non-zero median n-gram gap is a strong indicator of accidental epanaphora.*

The last numeric attribute to be considered is the length of the constituent sentences of an instance of epanaphora. Figure 5.38 shows how intentional epanaphora favour short sentence lengths of 15 tokens of less. However, in keeping in line with the previous decisions on boundaries for attribute value-based pruning we will be reducing this margin further to sentences whose length is less than ten tokens. This specific value was obtained by performing the third order polynomial interpolation as shown in Equation 5.1 on the datapoints from Figure 5.38.

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \tag{5.1}$$

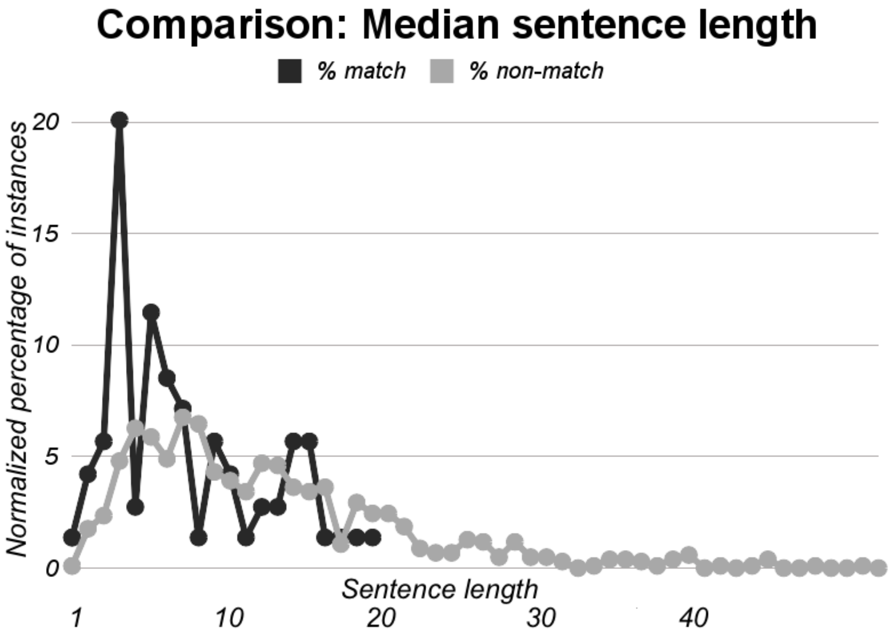


Figure 5.38: Annotation sentence length statistics

The resulting plot is shown in Figure 5.39. The actual values for the parameters are displayed in Figure 5.40. Figure 5.39 clearly shows the intersection of the two plots at a median sentence length of ten, making it a suitable cutoff value for the pruning of ambiguous instances of epanaphora. This value is actually fairly conservative.

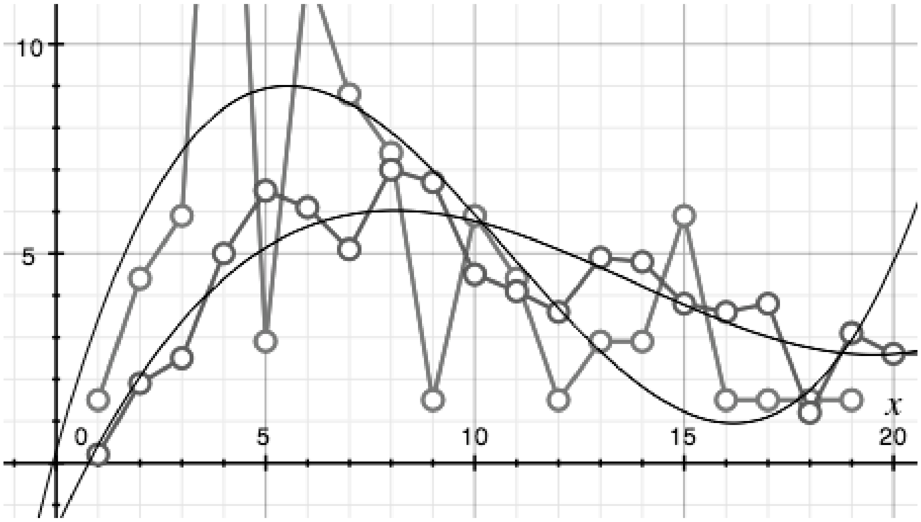


Figure 5.39: Annotation sentence length statistics with polynomial interpolation curves

Parameter	Value (match)	Uncertainty (match)	Value (nomatch)	Uncertainty (nomatch)
a_0	0.2787	±1.1333	-1.5763	±1.0217
a_1	3.564	±0.4782	2.1807	±0.3762
a_2	-0.4329	±0.0548	-0.1906	±0.0376
a_3	0.0133	±0.0018	0.0046	±0.0011

Figure 5.40: Polynomial interpolation values

Such a statement can be backed by plotting the normal distribution for the values in Figure 5.38 as per Equation 5.2.

$$y = e^{ax^2+bx} \tag{5.2}$$

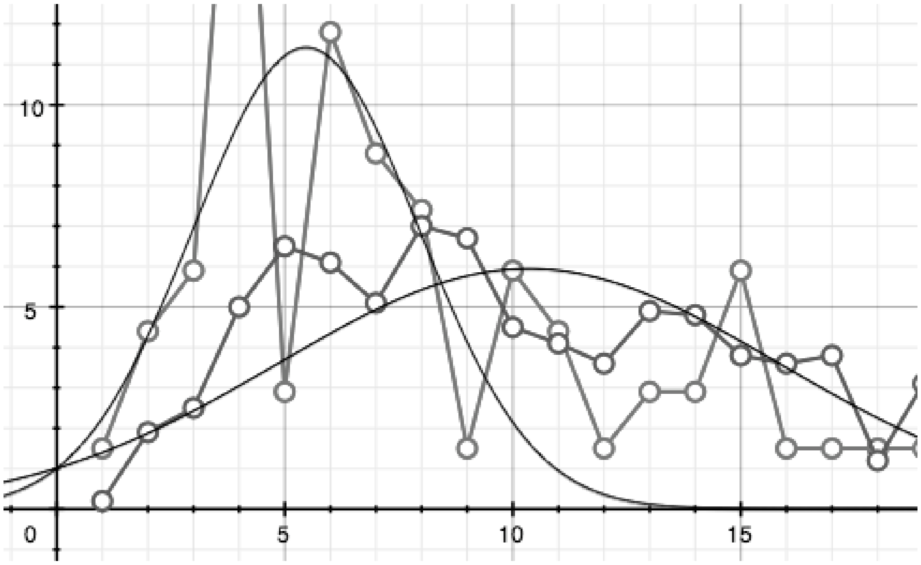


Figure 5.41: Annotation sentence length statistics with normal distribution interpolation curves

Parameter	Value (match)	Uncertainty (match)	Value (nomatch)	Uncertainty (nomatch)
a	-0.0814	±0.0045	-0.0166	±0.0014
b	0.8904	±0.0302	0.3441	±0.019

Figure 5.42: Normal distribution interpolation values

The normal distribution graph in Figure 5.41 places the cutoff value for pruning of ambiguous epanaphora at sentences of a length of eight tokens.

Observation 5.1.6. *Epanaphora whose constituent sentences have a median length of ten or more are likely to be accidental.*

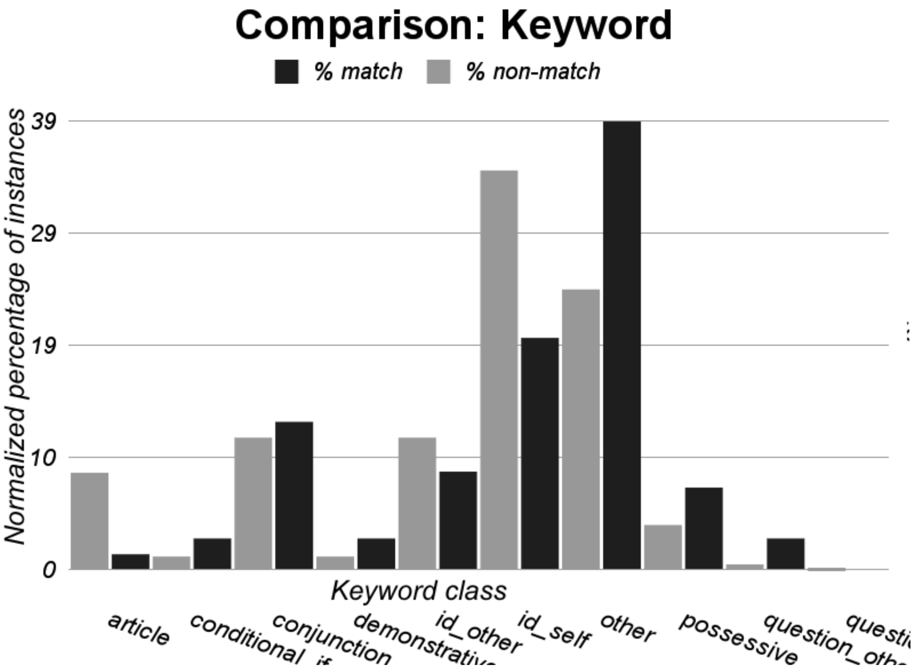


Figure 5.43: Annotation keyword statistics

The last epanaphora pruning criteria we are introducing is based on the keyword analysis performed during the annotation phase in Section 5.1.1. We had identified four categories in which the ratio of intentional to accidental epanaphora was disproportionate: 'Articles', 'reference of identity' (first person singular), 'questions', and 'other'. Of those four, only two are useful for generating pruning criteria, namely ‘articles’ and ‘reference of identity’. ‘Other’ and ’question’ are categories that place a higher emphasis on intentional epanaphora. As with the tuple width attribute these two categories would be used to enforce a classification of an instance of epanaphora as intentional, instead of dismissing a classification as accidental. The ‘article’ category is overwhelmingly accidental. This comes as no surprise, since articles are very generic tokens which occur with high frequency at the beginning of sentences. We can state with good confidence margins that the majority of the ‘article’ class instances of accidental epanaphora fall within the pruning criteria for epanaphora of n-gram length of one in Figure 5.34. The ‘reference of identity’ category, on the other hand, was not expected to be a candidate for the creation of pruning constraints. The reason for the high incidence of accidental epanaphora in this category can be attributed to the abnormal high frequency of recounts in singular first person in the input corpus. As a result we have stumbled upon the first corpus-specific constraint. We do not expect however that the inclusion or exclusion of this constraint should have too high of an effect on the quality of epanaphora classification. Unlike the ‘article’ keyword category which was unambiguously accidental, the first-person singular ‘reference of identity’ category is not overwhelmingly one-sided.

Observation 5.1.7. *Only the ‘article’ keyword category is a good constraint for keyword-based pruning.*

Constraint Testing

In order to test the effectiveness of the application of constraints we ran a new annotation sequence. The procedure was equal to that of the annotation process in Section 5.1.1 save for two items. First, all annotators received disjoint datasets. We are no longer focused on obtaining inter-annotator agreement statistics, which allows us to allocate independent input corpora to each annotator. The use of disjoint datasets in turn allows us to create a training set of equal size to the one from Section 5.1.1 at a faster pace and with less resource consumption. The second alteration was that instead of indiscriminately passing the input to the annotators, all the instances of epanaphora were pre-processed. Any instance of epanaphora that did not fall within the constraints for possible intentional epanaphora in Section 5.1.4 was automatically tagged as being accidental by the preprocessor and ignored by the annotation tool. The effect of this was two-fold: First, it allowed annotators to process the corpus at a much faster pace, since much of the tedium of repetitively encountering large blocks of epanaphora without a single candidate for intentional epanaphora was eliminated. Secondly the ratio of perceived instances of intentional epanaphora to instances of accidental epanaphora by the annotator was evened out. Table 5.44 reflects these results. Out of a total of 14444 instances of epanaphora from the input corpus a mere 1.025 percent met the constraint criteria for intentional epanaphora. This value actually comes close to the 1.282 percent rate for unanimous intentional epanaphora in Figure 5.4. Those instances of epanaphora which passed the constraint preprocessor were split nearly evenly among intentional and accidental during the actual annotation process, with 52.027 percent having been annotated as accidental and 47.973 percent annotated as intentional. In the pilot test in Section 5.1.3 the annotator-agreement corpus had been used to evaluate the accuracy of the classification algorithms. Since we have forfeit the generation of such a corpus

Description	Instances	Percent
Total accidental (automated)	14296	98.975
Total accidental (annotated)	77	0.533
Total intentional (annotated)	71	0.4916
Absolute total	14444	100
Training accidental (automated)	10076	98.94
Training accidental (annotated)	56	0.55
Training intentional (annotated)	52	0.511
Training total	10184	100
Test accidental (automated)	4180	99.052
Test accidental (annotated)	21	0.498
Test intentional (annotated)	19	0.45
Test total	4220	100

Figure 5.44: Statistics on constraint-based training and test corpora

for the constraint testing phase we needed to generate a different test corpus. For this purpose we have split the constraint-testing corpus into two separate corpora: One corpus for classifier training, and one for classifier evaluation. As can be seen in Figure 5.44 these two sub-corpora maintain the ratios of the parent corpus. This type of corpus is specially useful in this situation because it evaluates the classification algorithms as they would perform in the wild on generic input corpora.

It was concluded in Section 5.1.3 that the naive Bayes classifier had the best performance with regards to tagging instances of intentional epanaphora. It is for that reason that it was selected as the primary classification algorithm for testing the application of constraints. Figure 5.46 shows a very good receiver operating characteristic (ROC) plot with a very large area ratio under the curve. However, a look at the statistics in Figure 5.45 shows that the shape of the ROC curve is largely or entirely due to the statistics for instances of epanaphora classified as accidental. The naive Bayes algorithm's tendency to err in favour of tagging instances of epanaphora

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.842	0.132	0.028	0.842	0.054	0.955	yes
0.868	0.158	0.999	0.868	0.929	0.955	no
0.868	0.158	0.995	0.868	0.925	0.955	avg

Figure 5.45: Naive Bayes accuracy by class (constraint corpus, split test)

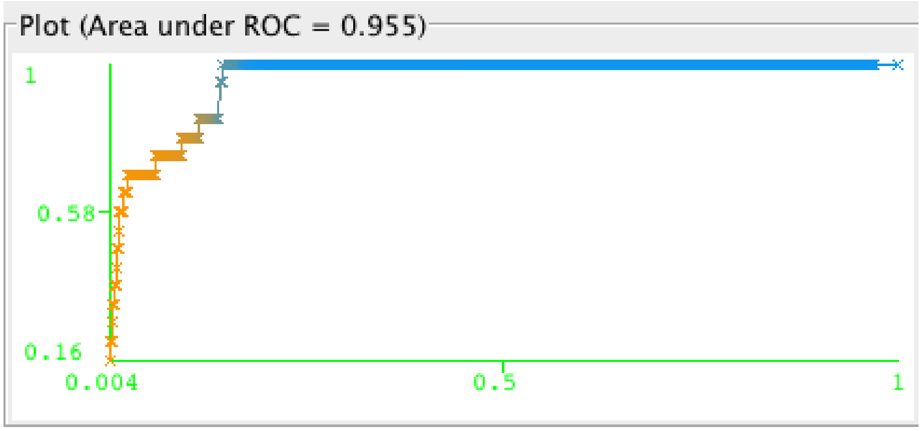


Figure 5.46: Receiver operating characteristic (ROC) curve for naive Bayes classification (constraint corpus, split test)

as intentional was welcome in light of the sparse training and test sets in Section 5.1.3 and perceived as a strength of the algorithm. However, in this section our goal has been moved from achieving high recall and precision to obtaining high precision values only.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0.159	0.057	1	0.108	0.972	yes
0.841	0	1	0.841	0.914	0.972	no
0.843	0.002	0.991	0.843	0.906	0.972	avg

Figure 5.47: Naive Bayes accuracy by class (constraint corpus without annotation, split test)

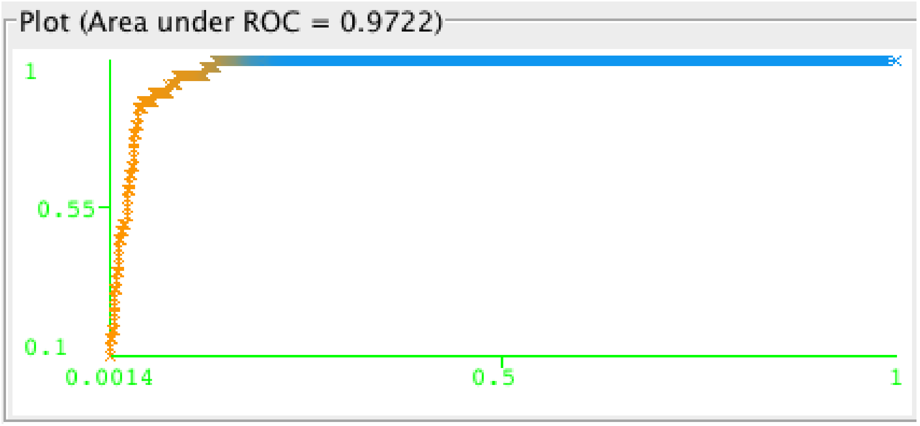

Plot (Area under ROC = 0.9722)

Figure 5.48: Receiver operating characteristic (ROC) curve for naive Bayes classification (constraint corpus without annotation, split test)

In Section 5.1.3 a change in the training corpus had brought forth better classification results during the algorithm test phase. In an attempt to replicate these results we created a variant of the training corpus. Instead of splitting the instances of epanaphora among intentional and accidental based on both constraint fitting and annotation we eliminated the latter for this corpus. All instances of epanaphora which were not rejected by the preprocessor were tagged as intentional, and the results were fed to the naive Bayes classifier. Figure 5.48 shows a better receiver operating characteristic curve, both in respect to the area under the curve as well as the shape of the curve itself. The precision measure for intentional epanaphora, as seen in Figure 5.47, also improved, effectively doubling when compared to the annotated corpus. This is a likely indicator that the Bayesian classifier is adjusting its parameters to mirror that of the preprocessor. However, it is still far outside the acceptable parameters for the precision rate.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.947	0.029	0.13	0.947	0.229	0.999	yes
0.971	0.053	1	0.971	0.985	0.996	no
0.971	0.053	0.996	0.971	0.982	0.996	avg

Figure 5.49: Non-naive Bayes accuracy by class (constraint corpus, split test)

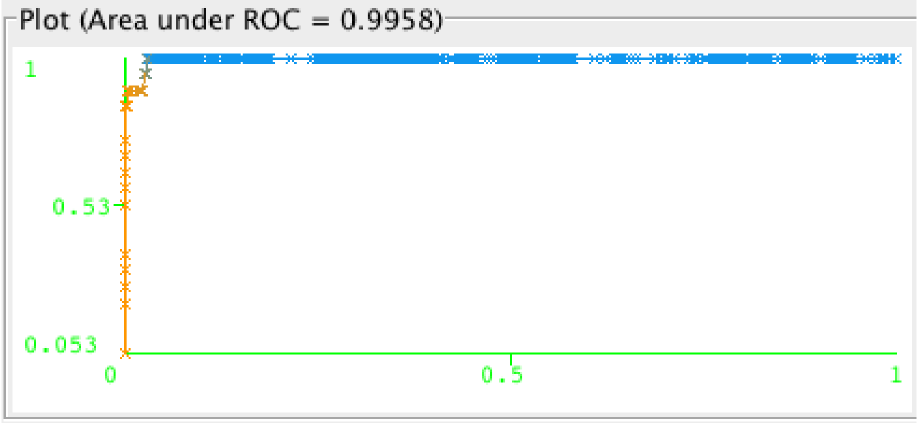


Figure 5.50: Receiver operating characteristic (ROC) curve for Bayes (non-naive) classification (constraint corpus, split test)

Despite the improved receiver operating characteristic curve and precision for the unannotated training corpus used for Figure 5.47 we felt that it was not the correct direction for the research goals in this section, and that instead it was only testing the classifiers' ability to imitate the constraint rules. The naive Bayes classifier had to be replaced, and the obvious strategy for this step was to test a different Bayesian classification algorithm. Figure 5.49 shows the results of training and testing a non-naive Bayes classifier with the same corpora that were used in Figure 5.45. The ratios resulting from the use of this classifier are much better than those of the naive Bayes classifier. However, a precision of only thirteen percent in classifying intentional instances of epanaphora is still too low, even with the improved ROC curve.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.842	0.002	0.64	0.842	0.727	0.999	yes
0.998	0.158	0.999	0.998	0.999	0.999	no
0.997	0.157	0.998	0.997	0.997	0.999	avg

Figure 5.51: M5P with regression accuracy by class (constraint corpus, split test)

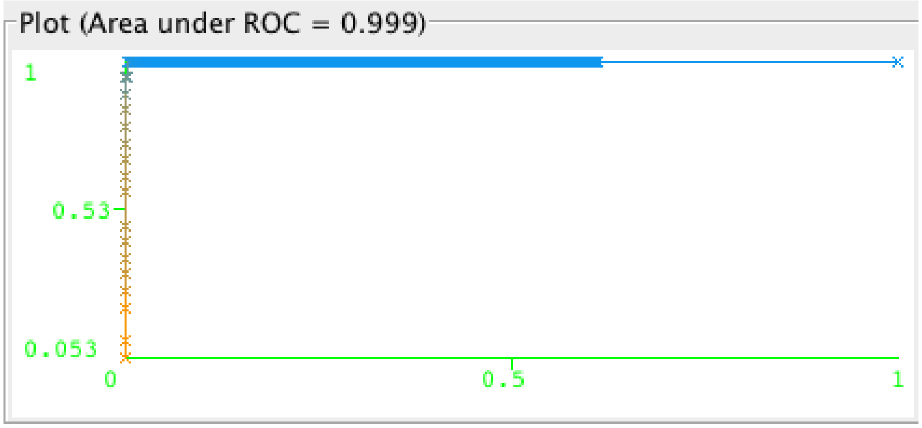


Figure 5.52: Receiver operating characteristic (ROC) curve for M5P with regression classification (constraint corpus, split test)

Having concluded that both the naive Bayes classifier as well as the non-naive Bayes classifier were unable to achieve a satisfying level of precision we began shifting our focus towards other algorithms. One of the most promising ones was the M5P decision tree algorithm combined with a classification via regression wrapper. This combination was capable of producing a near-perfect receiver operating characteristic curve (Figure 5.52) with a medium precision in classification of intentional epanaphora at a ratio of 0.64 (Figure 5.51).

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.789	0.002	0.682	0.89	0.732	0.999	yes
0.998	0.211	0.999	0.998	0.999	0.999	no
0.997	0.21	0.998	0.997	0.997	0.999	avg

Figure 5.53: Grafted J48 decision tree accuracy by class (constraint corpus, split test)

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.421	0.001	0.727	0.421	0.533	0.71	yes
0.999	0.579	0.997	0.999	0.998	0.71	no
0.997	0.576	0.996	0.997	0.996	0.71	avg

Figure 5.54: Nearest neighbour accuracy by class (constraint corpus, split test)

Other classification algorithms were explored with various results. J48 decision trees did not perform as well as M5P trees with the regression wrapper, but grafted J48 trees managed to eke out a small advantage in precision, as can be seen in Figure 5.53. The nearest neighbour like algorithm using non-nested generalized exemplars (NNge) in Figure 5.54 had the best precision, but at the cost of all other statistics, making it likely that the good precision measure may have been a fluke. These results put the decision tree algorithms above all others, with grafter J48 trees having producing the best results, followed very closely by M5P trees with a regression wrapper. Bayesian classifiers took the bottom with the worst precision performance. These results are a complete reversal from those in Section 5.1.3. A possible explanation for this divergence is the different size for the training and test corpora used. The datasets used in Section 5.1.3 were very small. The constraint preprocessor enabled us to generate larger datasets, addressing one of the major reasons for the poor performance of the other classification algorithms.

5.2 Study Expansion - New Corpus

Throughout Section 5.1 the nucleus of the research has revolved around the use and study of the corpus identified in Section 3.1.2, namely the 2006 TREC Blog Track. The justification for this focus had been that the TREC Blog track is a good representative of contemporary prose, domain-independent, and is structured in a sufficiently predictable way to allow easy paragraph and sentence boundary detection. However, focusing on one corpus alone does not give us the opportunity to test the robustness of the epanaphora classification rules when applying them to a different literary domain. To address this shortcoming we introduced a new corpus to our research. The new corpus is sourced from late 19th Century Canadian literature in the form of printed and published texts which were fed through an Optical Character Recognition (OCR) system to convert them to a generic machine-readable digital format. This corpus was chosen for a variety of reasons. It is contemporary enough to maintain a vocabulary similar to the TREC Blog corpus. This choice was made in order to facilitate the reuse of the sentence boundary detection algorithm. By choosing corpora which were written with similar vocabularies we hoped to minimise the need to rebuild the abbreviation dictionaries for the new corpus. The second criteria for selecting the Canadian literature corpus was the desire to focus more on edited texts with complete sentence structures. A large quantity of the content from the TREC Blog corpus was composed in a personal writing style. Such style is not designed for extensive sentence planning and revision, which impedes the epanaphora classification process in that it increases the work required to produce a balanced training corpus. We reason that the repetitions of n-grams encountered in the Canadian literature corpus are more likely to be of an intentional nature.

5.2.1 Corpus Preparation

The first task in preparing the Canadian literature corpus for epanaphora detection and classification was to convert it to a format compatible with out annotation framework and classification schemas. Fortunately the OCR system already produced machine-readable documents. However, these documents contained a large amount of extraneous metadata. We designed a preprocessor to extract the text content from these documents and store it in our own existing file format using Extensible Markup Language (XML).

Following that we proceeded with the epanaphora detection process from Section 4.3. The output of this routine was our new base corpus. It contains the largest possible set of unique instances of detected epanaphora for the Canadian literature corpus, and is ready to be used by the epanaphora classification system. However, we decided to postpone that step and perform additional refinements on the algorithm before proceeding with the classification routines. Specifically we chose to apply the pruning mechanism from Section 5.1.4. We were sufficiently confident that the constraints that were defined in that segment of the study are adequate for application to any epanaphora classification process whose goal is to solely detect intentional instances of epanaphora. In fact, after a review of the pruning criteria we decided to further tighten the constraints. For the attribute sets relating to n-gram length, tuple gaps, and sentence length we had initially used only the median values, as opposed to any of the maximum, minimum, or average values available. However, when working with the new corpus we decided to apply tighter restrictions to the attributes based on the nature of the constraints laid on them.

For n-gram lengths the applied constraints are intended to place a lower bound on the n-gram length of an instance of epanaphora before it is allowed to be tagged as intentional. It is therefore more appropriate to use the minimum n-gram length among a set of sentences as the discriminating value instead of the median n-gram length. In addition of placing a harder limit on the minimum n-gram length of an instance of epanaphora this constraint synchronises the design of the pruning mechanism with that of the epanaphora detection algorithms in Section 4.3.3, since they themselves use the minimum n-gram length as a bound for recursion.

The use of attribute values for the tuple gap sets was also changes for the same motivation which led to the change in use of n-gram length attributes. In Section 4.3.3 the maximum gap size between constituent sentences of an instance of epanaphora was used as an additional method for producing epanaphora variants as per Algorithm 4.6. It therefore stands to reason that the maximum gap size allowed should replace the median tuple gap size allowed for the epanaphora pruning process.

An additional side-effect of switching the n-gram length constraint to be equal to the lower bound and the tuple gap constraint to be equal to the upper bound is that two of the attributes used by the pruning system are now reduced to dealing with absolute values instead of approximations.

Lastly we altered the way in which keyword categories were used as constraints within the pruning mechanism. In Section 5.1.4 we had dismissed the *article* keyword category as being largely accidental. However, we later speculated that the reason for this classification is not specifically due to the sentences beginning with the first token of the incumbent sentences being identified as articles, but due to the fact that articles are too generic as keywords. They are context-free *filler* words, and as such carry little weight by themselves. Their occurrence in instances of epanaphora needs to be taken into account when considering the n-gram length attributes. We modified our constraint application process to consider co-occurrences of attributes. As a result we were capable of relaxing the constraints for instances of epanaphora tagged with the *article* category. Instead of rejecting all instances of epanaphora belonging to this keyword category we introduced additional rules requiring a larger lower bound for the n-gram length attribute. The result is equivalent to the first word of sentences beginning with article keywords is considered to be a zero-length token.

Two more keyword categories were introduced to the pruning algorithm. The first was the use of conditional tokens, i.e. the *conditional* keyword category. This category was treated like the *article* one by treating the first word of constituent sentences as a a zero-length token for the purpose of n-gram length analysis. The second category added to the constraint procedure was the one tagging conjunctions. In the TREC Blog track, sentences beginning with these keywords were allowed by default. For the Canadian literature corpus we reversed that choice. We justify this decision by pointing out that repeated sentences beginning with conjunctions are not grammatically complete. While their occurrence was permitted in the TREC Blog corpus due to the prevalence of the personal writing style, we decided that the instances of intentional epanaphora from the Canadian literature corpus should strive to contain only well-formed sentences.

5.2.2 Domain Analysis

The classification procedure that was prevalent throughout Section 5.1 involved the human-assisted creation of training, classification, and test datasets. The strategy behind the generation of such datasets was to address the steps necessary to evaluate various machine learning algorithms and test them for performance. Under such circumstances it was fundamental to have standardised datasets throughout each step in order to facilitate comparing and contrasting of various algorithms. However, at the current stage of this thesis' research such standardised datasets are no longer compulsory.

The first task in performing classification on the new corpus was to determine the quality of the constraints applied by the pruning algorithm itself. We ran the M5p decision tree with regression classification – the preferred classifier from Section 5.1.4 – against the pruned corpus. Every pruned instance of epanaphora was tagged

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.046	0.006	0.399	0.046	0.082	0.928	yes
0.994	0.954	0.928	0.994	0.96	0.928	no
0.924	0.884	0.889	0.924	0.895	0.926	avg

Figure 5.55: M5P with regression accuracy by class (literature corpus)

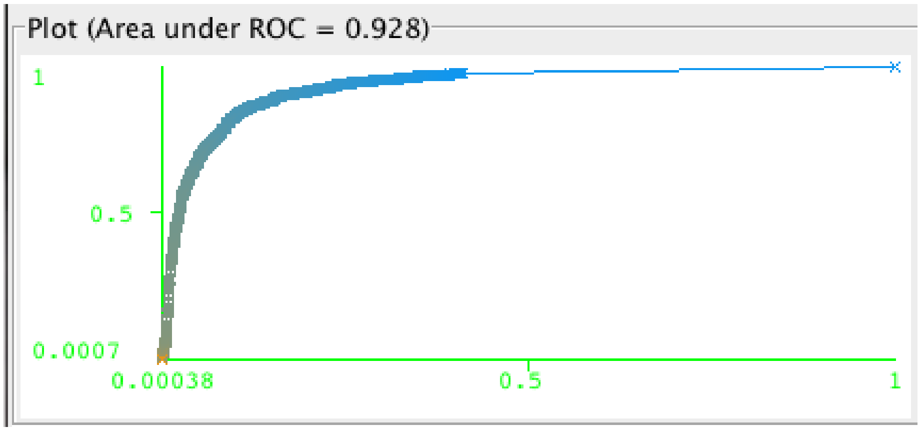


Figure 5.56: Receiver operating characteristic (ROC) curve for M5P with regression classification (literature corpus)

as being accidental, and the rest was tagged as intentional instances of epanaphora. The ROC plot shows a a sizeable area under the curve in Figure 5.56, hinting at a fairly reliable classification result. However, the shape of the curve is significantly more shallow than those from Section 5.1.4. Looking at the ratio table in Figure 5.55 we can see the source of this reversion: The recall measure for intentional instances of epanaphora is extremely low. By further examining the confusion matrix for the classifier we further see that out of a total of 7116 instances of epanaphora that were tagged as intentional – that is, instances of epanaphora that passed the constraint rules – less than five percent (324 instances) were classified by the algorithm as being intentional. By contrast, only 0.551 percent of the tagged instances of accidental epanaphora were classified as being intentional. What this tells us is that the constraints that were applied to the TREC Blog corpus are too loose for the Canadian literature corpus. To verify this we had a human annotator process a sample of the set of constraint-approved instances of epanaphora. The results were quite telling. Figure 5.58 shows nearly twice as many instances of accidental

	Classified	
Tagged	intentional	accidental
intentional	324	6792
accidental	488	88079

Figure 5.57: Confusion matrix: M5P with regression accuracy by class

	Instances	Percent
Total	223	100
Intentional	77	34.529
Accidental	146	65.471

Figure 5.58: Annotated epanaphora distribution

epanaphora than instances of intentional epanaphora. This greatly differs from the ratios in Section 5.1.4, where the instances of epanaphora were split evenly among the intentional and accidental categories. It is also evident that the previous hypothesis about insufficient pruning holds true. Delving further into the distribution by attribute shows nothing out of the ordinary for numeric attributes, but the distribution of epanaphora by keyword category reveals the likely source of the observed reduction in accuracy. Figure 5.59 shows the normalised distribution of epanaphora per keyword class. While sporting a scattering that diverges somewhat from that of the original TREC Blog corpus (Figure 5.10) it does still demonstrate a healthy spread among the various keyword categories. In contrast the instances of epanaphora in the Canadian literature corpus are almost exclusively allotted into the *other* keyword category. We take this as evidence of a need to implement a more atomic form of keyword classification for domain-specific texts. Before beginning with that area of research, however, we need to determine a baseline accuracy that can be used as a starting point for new developments.

The table in figure 5.61 shows exactly how endemic the issues are that arise from a lack of keyword categorisation. While the algorithm was capable of correctly identifying all annotated instances of intentional epanaphora, it was unable to discriminate the majority of annotated accidental instances of epanaphora. The algorithm's inability to to use the keyword attribute for classification is also compounded by a an already-tightened variety for the attributes which were involved in pruning. In the end, all those complications come together to produce a receiver operating characteristic curve (Figure 5.62) which hardly deviates from the diagonal.

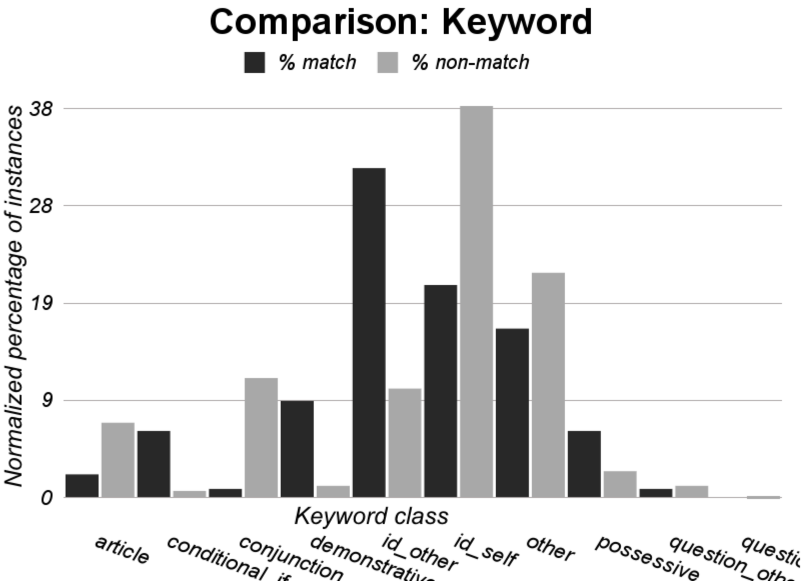


Figure 5.59: Training corpus distribution by keyword

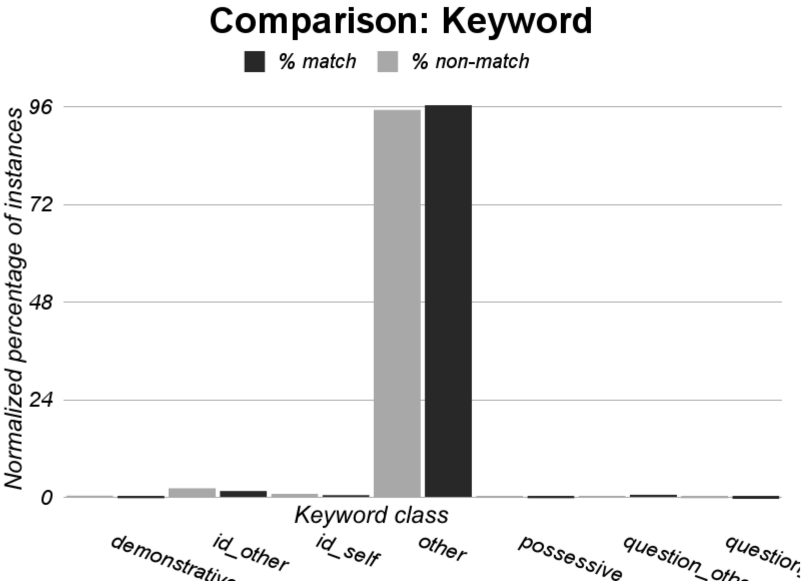


Figure 5.60: Test corpus distribution by keyword

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0.774	0.405	1	0.577	0.575	yes
0.226	0	1	0.226	0.369	0.575	no
0.493	0.367	0.795	0.493	0.441	0.575	avg

Figure 5.61: M5P with regression accuracy by class (literature corpus, constrained)

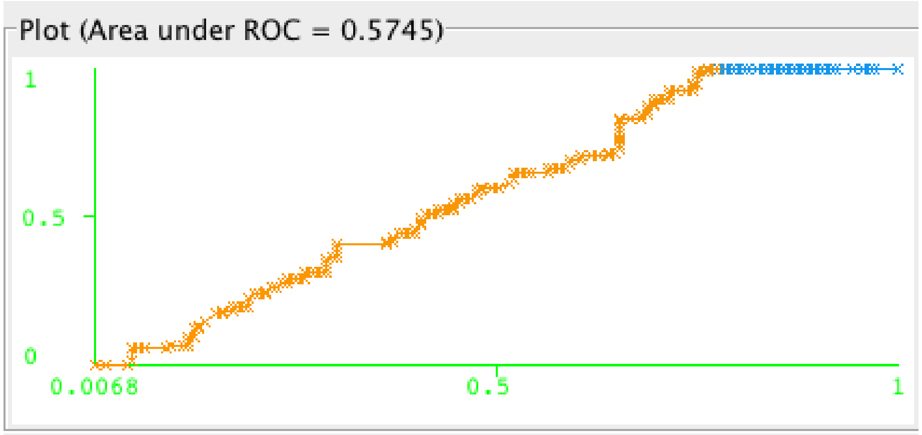


Figure 5.62: Receiver operating characteristic (ROC) curve for M5P with regression classification (literature corpus, constrained)

Since the machine learning algorithms are incapable of performing sufficiently accurate epanaphora classification with their current training set we need to find a different approach to improve the accuracy of the epanaphora classifiers. A simple approach is to return to a more hands-on approach to examining the Canadian literature corpus.

5.2.3 Corpus Characteristic Re-Evaluation

Having already discovered that the constraints applied during the pruning process are insufficiently tight we decided that it had become necessary to scrutinise the instances of epanaphora for context information.

Accidental Epanaphora

We had come to the conclusion that the first observation task should be to identify the types of accidental epanaphora which do not get caught by the constraint application algorithm. Two types of repetition are immediately apparent. One of these types is *lists*, and the other one is *legal documents*.

...
Chief Events of *Grecian History.*
Chief Events of *Roman History.*
Chief Events of *Eastern Empire.*
Chief Events of *German History.*
Chief Events of *English History.*
...

Figure 5.63: Excerpt from the table of contents of a history book

It is easy to understand why list-type repetitions are difficult to exclude from the list of candidates for intentional epanaphora. These kinds of repetitions are commonly composed of multiple short sentences with very similar grammatical structure. They therefore fall within the acceptable bounds for all the primary constraints: The sentences are not excessively long, they are consecutive (gapless), they regularly contain repetitions involving three or more sentences, and the list format usually guarantees that many of the involved list items (sentences) will follow the same overall structure, increasing the incidence of longer n-grams.

Knowing that these types of repetitions fall within the given constraints despite being almost entirely accidental, how can they be eliminated from the set of candidate instances for intentional epanaphora. To answer that question we were required to dig deeper into the texts and their content. Continued examination of the list-type repetitions yielded a keen insight into their nature. We discovered that these types of repetitions are most commonly the result of the OCR software indiscriminately parsing all pages of a printed text, including indices, tables of content, and data tables with repeating headings. While this is usually not a significant issue, the relevant context information was lost when converting the output of the OCR software to a format compatible with the epanaphora detection and classification software. The result is that most, if not all, of the indicators which would have been capable of identifying these portions of text as irrelevant were discarded during the format conversion process. We conclude that it is therefore necessary to pay a higher degree of attention to the structure of a document before parsing it for epanaphora detection and classification.

Throughout the examination of the texts in the Canadian literature corpus we discovered that the documents it contained were not only a compilation of literary

. . . Gold Medal for Oil Painting, Miss Carrier, New York. ***Silver Medal for*** *Oil Painting, Miss Mitchell, Ottawa.* ***Silver Medal for*** *Crayon Drawing, Miss Maggie Dowdall, Merrickville.* ***Silver Medal for*** *Drawing in Coloured Crayons, Miss M. E. Kelly, Grenville . . .*

Figure 5.64: Excerpt from an awards list

. . . Ten livres of small glass Beads – white, green,and transparent. ***One gross of*** *large Clasp-knives, with horn handles.* ***One gross of*** *round buckles, both large and medium-sized.* ***One gross of*** *small metal plates. . . .*

Figure 5.65: Excerpt from an inventory

texts, but that they also included texts of legal nature such as court and legislative proceedings, sales inventories, land ownership declarations, and other types of records. What all these texts have in common is that they are records, and none of them are written with any rhetorical function in mind. They are clear examples of the domain-sensitive nature of epanaphora classification, and indicators that it is not wise to have this type of heterogeneity in a corpus.

Special Categories of Intentional Epanaphora

Until this point we have focused our examination on the attributes of the instances of accidental epanaphora which have eluded pruning through the application of constraints. We discovered two types of categories of text to which the greatest amount of instances of accidental epanaphora could be attributed. Section 5.2.3 first focused on indices and tables of content attached to literary documents. It then proceeded to put the spotlight on the heterogeneous nature of the input corpus. However, no evaluation has yet been made as to the qualities of the instances of epanaphora which

> *. . . One copy to the Lieutenant-Governor. Six copies to the Legislative Council. Two copies to the Executive Council.* ***One copy to*** *each Member of the Legislative Council.* ***One copy to*** *each Member of the Executive Council.* ***One copy to*** *the Chief Justice.* ***One copy to*** *the Master of the Rolls. . . .*

Figure 5.66: Excerpt from the proceedings of a council meeting

were annotated as being intentional.

Two groups of instances of intentional epanaphora ended up coming to our attention. One such group was that of rhetorical questions. These types of rhetorical figures are very prominent within their context, and they often occur in clusters. The second category of intentional epanaphora came from texts with high emotional appeal. Among these, two common sub-categories stand out: Religious texts, and patriotic references.

> ***Has she*** *loved you as I have done?* ***Has she*** *humbled and made so little of herself as to tell you so, as I have done?* ***Has she*** *helped you into a paying practice, introduced you to society, knowing your dark secret all the time?*

Figure 5.67: Example of a cluster of rhetorical questions starting with a verb

What do I care? What do I care? What do I care?

Figure 5.68: Example of a repeating question

The sub-category of rhetorical questions was unexpected, given the low incidence of question keywords shown in Figure 5.60. However, it soon became apparent why these instances of epanaphora were not being picked up by the keyword categoriser. Most of the rhetorical questions encountered during the annotation phase of the Canadian literature corpus did not contain a question keyword as the first token of their constituent sentences, but a verb instead. Figure 5.67 shows an example of such a cluster of rhetorical questions. This is yet another indicator that the keyword categorisation is flawed, or at the very least insufficiently atomic.

Have mercy! Have mercy! Have mercy!

Figure 5.69: Example of a repeating exclamation

Hurrah for *love!* ***Hurrah for*** *hope!* ***Hurrah for*** *industry!*
Hurrah for *bonnie Canada, And her-bonnie maple tree!*

Figure 5.70: Example of a cluster of patriotic exclamations

As was mentioned earlier, two categories of the intentional epanaphora with high emotional content stand out: Religious texts, and patriotic references. The former category, of which an example can be seen in Figure 5.69, has a strong emphasis on short, repeated exclamations. These are frequently linked to church songs an chants, and have a format which encourages simplicity. The second sub-category, patriotic references, is more interesting in that it frequently occurs in poetic sources, such as the poems of Thomas D'Arcy McGee [45] and The Emigrant's Song [40]. The high emotional content of these poems can very well be justified by considering the historical context of their authorship.

A side-by-side examination of the two categories discussed above – rhetorical questions and texts with high emotional content – reveals further parallels between the two. First there is the punctuation used. Rhetorical questions, by their very nature, are terminated with a question mark. The instances of epanaphora with high emotional content, on the other hand, are not required to be terminated with any particular symbol. However, a prominent portion of those types of intentional epanaphora are terminated with an exclamation mark. We therefore postulate that the type of punctuation used among the constituent sentences of an instance of epanaphora can be used to discriminate certain types of intentional epanaphora.

The second parallelism is the verbatim repetition of the same sentence, as seen in Figures 5.68 and 5.70. These types of rhetorical figure feature a strong correlation between the numeric values of their attributes. In the most obvious cases, this correlation is that the minimum n-gram length η_{min}, the maximum n-gram length η_{max}, and the minimum and maximum sentence length all have the same value. Further variations can be introduced by loosening the constraints as needed. Finally, this particular correlation between the values of the numeric attributes for the instances of epanaphora exposes a shortcoming of the attribute format and its role in the training of the epanaphora classification algorithms. The format in which the attributes were recorded does not provide any explicit relation between the attributes, their values, and the values of other attributes. It is therefore impossible for example for a classification algorithm to know that the values of the minimum and maximum n-gram lengths for any instance of epanaphora must be no less than the minimum sentence length and no more than the maximum sentence length for that same instance of epanaphora.

Lastly there is the matter of keyword categories. It is already evident that the primary task for improving this attribute is the need to generate a useful subdivision of the *other* category. Given the revelation that many of the intentional instances of epanaphora from clusters of rhetorical questions constitute of groups of sentences beginning with verbs we propose that part of speech tagging should be used as one of the criteria for subdivision.

5.2.4 Analysis of Effectiveness of Special Categories

In Section 5.2.3 we had identified two types of categories of intentional epanaphora with special traits. These were those with special termination punctuation – question marks and exclamation signs – and those with a strong correlation between n-gram length and sentence length. In this section we will examine the results from applying additional meta-constraints to the candidate instances of epanaphora.

Punctuation

	Instances	Percent
Total	110	100
Intentional	101	91.818
Accidental	9	8.182

Figure 5.71: Sample annotated epanaphora distribution (question terminator)

	Instances	Percent
Total	81	100
Intentional	80	98.765
Accidental	1	1.235

Figure 5.72: Annotated epanaphora distribution (exclamation terminator)

The values for the ratio in Figures 5.71 and 5.71 are telltale signs of the effectiveness of introducing punctuation terminators as criteria for automatic classification of instances of epanaphora. Both the question marks as well as the exclamation signs are extremely effective at selecting a high ratio of intentional epanaphora. The question

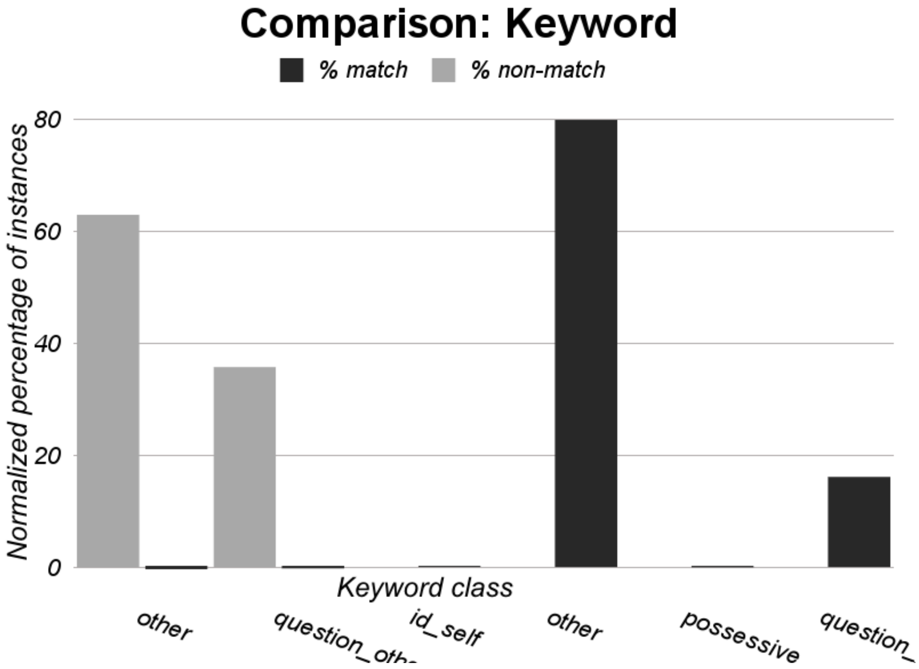


Figure 5.73: Test corpus distribution by keyword (question terminator)

terminator in particular displays a marked distribution of instances of epanaphora among keyword categories. Figure 5.73 shows that the marked rhetorical questions are more likely to be tagged as *other*, reinforcing the idea from Section 5.2.3 that part of speech categorisation will prove to be a vital expansion of the keyword attribute for instances of epanaphora.

The highly intentional nature of both rhetorical questions and epanaphora which are terminated with exclamation signs supports the argument from Section 2.3.1 that the appeal to pathos plays a significant role in rhetorical figures. We can therefore conclude that being able to determine whether an instance of epanaphora has a high pathos appeal is a strong factor in distinguishing between intentional and accidental epanaphora.

Attribute Homogeneity

	Instances	Percent
Total	130	100
Intentional	30	23.077
Accidental	100	76.923

Figure 5.74: Sample annotated epanaphora distribution (homogeneous sentence length)

	Instances	Percent
Total	32	100
Intentional	31	96.875
Accidental	1	3.125

Figure 5.75: Annotated epanaphora distribution (homogeneous sentence length, list entries removed)

Just as the application of sentence terminator constraints proved to exceed all expectations of effectiveness, so did classification by attribute homogeneity fail to produce a satisfactory ratio of intentional to accidental epanaphora. The attempt to use matching values for the sentence length and n-gram length attributes backfired. Instead of returning a subset of intentional epanaphora with short and concise repetition, the type of constraint applied by eliminating sentences with incongruous sentence lengths revealed a higher ratio of list-type repetitions. The manifestation of these types of repetitions has again proven to have a major negative effect on our system's ability to identify instances of intentional epanaphora in an unsupervised fashion. However, Figure 5.75 shows that after manually eliminating all accidental entries that were part of a list or table the results were just as good as those encountered when constraining candidate instances of intentional epanaphora to sentences with specific terminators. What we can take away from this is that the careful application of constraints to the input corpus can help improve the quality of detection and classification of intentional epanaphora.

5.3 Summary of Results

Based on the results of our study, we recommend that the approach to detecting and classifying intentional epanaphora should be treated as a constraint satisfaction problem. We recommend two sets of constraints, one set to eliminate those repetitions which have the greatest likelihood of being accidental, and one set to select the remaining repetitions which have the greatest likelihood of being intentional. Our results show that the elimination set of constraints should include the following steps:

- Single-token n-grams (n-grams of length one) are primarily accidental and should be discarded.
- Sentence clusters beginning with article keywords should be discarded. This rule is of particular importance for repetitions with short n-gram length. Articles have a much lower semantic significance than all other word classes, and as a result introduce too much noise to our measurements.
- The likelihood of intentionality drops exponentially with gap width. We recommend not to allow a median gap width of more than one sentence between the constituents of a repetition.
- Long-sentence repetitions are primarily accidental according to our results. We recommend using an initial sentence length limit of ten or more words, followed by a tuning of this variable in order to maximise the f-measure after applying this constraint.

The set of selection constraints for classifying instances of epanaphora as intentional is significantly shorter than the set of elimination constraints. This is not an indicator of a lack of intentionality selectors, but a testament to the efficiency with which the set of elimination constraints is capable of trimming the list of candidate instances of intentional epanaphora, as well as the effectiveness of the selection constraints in selecting clearly intentional instances of epanaphora. These constraints are as follows:

- Punctuation is the most important selector for intentional epanaphora. Sentence clusters which are terminated with exclamation signs and question marks are particularly likely to be part of an intentional instance of epanaphora, even without the application of the elimination constraints. This property of punctuation makes it a key attribute for the selection of intentional epanaphora.

- The one secondary attribute which can be used to detect instances of intentional epanaphora is the homogeneity of sentence length. Sentence clusters with low sentence length deviation have a greater likelihood of forming an instance of intentional epanaphora. However, this attribute is applicable only to sentences which are not part of a special class, namely list entries. Lists should be treated separately in this case because their length is constrained by convention.

One attribute which is notoriously absent in these sets of elimination and selection constraints is the tuple width of intentional epanaphora candidates. There is no uniquely distinguishing tuple width value which can be used reliably to place a candidate instance of epanaphora in either the intentional or accidental group.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Throughout this thesis we have exposed the richness of knowledge that is encased by rhetorical figures. We have furthermore shown that this knowledge can be exploited via computational methods to establish novel methods of document analysis.

6.1.1 Epanaphora Detection and Classification

We chose epanaphora as the rhetorical figure focus for our study. As a figure of syntax (schemes) epanaphora was an ideal candidate for this research. Its syntactic nature made its detection not only easy, but it also allowed us to minimise the incidence of errors by not having to rely on possibly inaccurate information such as automated part of speech tags. Despite its syntactic simplicity epanaphora has proven to contain a wealth of research aspects. We guided our process of discovery towards two major categories of epanaphora: Accidental and intentional. Of these two categories, we focused on the detection and classification of intentional epanaphora. The detection process was split from the classification process, which allowed us to tune their parameters individually. Detection of epanaphora was performed on a global basis, including both accidental and intentional epanaphora, thus allowing us to guarantee a comprehensive set of results. This comprehensive set is then used by the classification aspect of our research. Since the input set for classification is guaranteed to be comprehensive we were able to limit the scope classification of the intent of epanaphora to a filtering process. Fine-tuning of epanaphora classification was thus achieved by modifying the parameters used to discriminate between accidental and intentional epanaphora.

We have shown that while the distinction between intentional and accidental epanaphora is evident to human observers when pointed out, automated classification between intentional and accidental epanaphora is not trivial. The accuracy with which the examined attributes of instances of epanaphora are capable of determining the intentional nature of epanaphora varies from one attribute to another. Furthermore the reliability of the classification by attributes depends greatly on the interaction between said attributes. For example the first word (keyword attribute) of the constituent sentences of an instance of epanaphora has a strong effect on the degree with which the n-gram length of an instance of epanaphora determines its intentional nature. Auxiliary keywords such as articles diminish an n-gram's accuracy in correctly classifying an instance of epanaphora as intentional or accidental, in particular for short n-grams. Despite these variances we have discovered that certain combinations of attributes are particularly effective at eliminating instances of accidental epanaphora. Among these combinations are short tuple gaps, wide tuples, short sentences, and medium n-gram lengths. We were able to use these combinations to generate a constraint-based pruning processor which is capable of balancing the ratio of detected intentional to accidental epanaphora in a corpus. We also discovered that certain types of punctuation – question marks and exclamation signs – play a significant role in the intentional nature of instances of epanaphora. An analysis of the text of the instances of intentional epanaphora falling into this category further revealed that their contents were laden with high emotional appeal. This reinforces the notion that pathos, the appeal to emotion, is a strong contributing factor in the intentional nature of epanaphora. Conversely, the strong emotional appeal in these rhetorical figures supplements the argument that repetition – which is the core concept of epanaphora – is a substantial contributing factor to saliency.

We also evaluated the most prominent instances of accidental epanaphora and concluded that context is one of the leading factors in determining the accidental nature of instances of epanaphora. The type of document, the epanaphora instance's location within its document, and the text's writing style (prose, poetry, non-narrative texts) heavily influence the accuracy of epanaphora classification.

6.1.2 Value of Epanaphora Detection and Classification

Through the work in our thesis on the detection and classification of epanaphora we have shown that rhetorical figures can be used as significant and reliable indicators for document analysis. We have further demonstrated that the shallow attributes of repetitions are sufficient to create a dependable set of rules for epanaphora detection and classification. This in turn shows that it is not necessary to perform deep analysis of text to extract information about author intent, but that it is more important to look at the appropriate set of attributes and couple them with a relevant set of rules. The templates for these rules are already available in the form of definitions for rhetorical figures. The work in our thesis is valuable in this aspect because it proves that it is possible to use the definitions of rhetorical figures to create computationally significant rules. In return we justified these definitions by providing empirical results which corroborate the general assertions implicit in the definition of epanaphora. As a result we contribute to the study of rhetoric itself by providing the tools and results necessary to disambiguate, formalise, and improve the definitions of rhetorical figures.

6.2 Future Work

As a new research field, the computational use of rhetorical figures in document analysis promises a wealth of research opportunities. In our thesis we have only barely opened the door to this field, and there are multiple directions in which this work can be taken in the future. However, there are some tasks which are more urgent than others.

We find that the most critical short-term research aspect of the computational study of epanaphora is the accuracy with which intentional epanaphora can be distinguished from accidental epanaphora. The correlation between the attributes of instances of epanaphora and their intentional or accidental nature is still largely occurring on an individual basis. We have shown that introducing constraints build upon the combined values of two or more attributes can improve a classifier's ability to distinguish between intentional and accidental epanaphora, but there are still many variants to explore. We have also shown conclusively that the keyword attribute is too immature to be an effective measure of intentionality. While the exclusion of keyword attributes from the list of selection constraints for intentional instances of epanaphora does not have a negative effect on our model's capability to discriminate between intentional and accidental instances of epanaphora, we suggest that other venues of classification can be introduced which can take place instead of simple keyword attributes. Among these venues are part of speech tagging, word frequency, and weighted discrimination of keyword attributes.

Finally we need to introduce more attributes, specifically those related to sentence context and text structure. We have shown that a prominent subsection of accidental epanaphora come from non-narrative texts such as indices, lists, and data tables, and that it is recommended that a distinction be made between narrative and non-narrative texts in order to improve the accuracy of epanaphora classification. We furthermore believe that the layout of a document is an additional surface attribute of said document whose correlation with rhetorical figures should be explored in further studies.

Beyond the possible accuracy improvements in epanaphora detection we also find that it is worthwhile to further explore the variations in intentional and accidental epanaphora from one text corpus to another. The opportunities in this area are endless. Historical variations, changes in narrative style by author, prose versus poetry, and even language can contribute to different interpretations of the intentional or accidental nature of instances of epanaphora.

We hope that future research will follow the trail laid out in this thesis and further explore the relation between detection and classification of intentional epanaphora and document classification. Having demonstrated that epanaphora is a valuable metric attribute itself, we wish to see how this attribute can be applied to existing metrics for document classification and analysis.

Lastly there are all the other rhetorical figures which can be studied. Rhetorical figures have been largely deemed to either be too inaccurate or to lack sufficient information density to be of use in computational linguistics. In this thesis we have demonstrated that with due diligence the introduction of rhetorical figures benefits not only computational linguistics but also other fields in computer science such as document classification and information retrieval. The crucial step however is to update the historical descriptions of each rhetorical figure with modern definitions that can be used to generate rulesets for the detection and classification of said figure.

Bibliography

[1] J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain. MITRE: description of the Alembic system used for MUC-6. In *Proceedings of the 6th conference on Message understanding*, pages 141–155. Association for Computational Linguistics Morristown, NJ, USA, 1995.

[2] D.E. Appelt. Planning English Sentences. 1992.

[3] Douglas E. Appelt. Planning english referring expressions. *Artificial Intelligence*, 26(1):1–33, 1985.

[4] H. Baayen, H. van Halteren, A. Neijt, and F. Tweedie. An experiment in authorship attribution. *Proc. JADT*, pages 69–75, 2002.

[5] H. Baayen, H. van Halteren, and F. Tweedie. Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–132, 1996.

[6] R.W. Bailey. Authorship attribution in a forensic setting. *Advances in Computer-Aided Literary and Linguistic Research*, 1979.

[7] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 805–810. Lawrence Erblaum Associates LTD, 2003.

[8] P. Baroni and M. Giacomin. A systematic classification of argumentation frameworks where semantics agree. In *Computational Models of Argument, Proceedings of COMMA*, volume 172, pages 37–48, 2008.

[9] R. Barzilay and M. Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

[10] B. Beigman Klebanov and E. Shamir. Reader-based exploration of lexical cohesion. *Language Resources and Evaluation*, 41(1):27–44, 2007.

[11] Bruce Bertram et al. Why readability formulas fail. reading education report no. 28. Technical report, National Institute of Education (ED), Washington, 1981.

[12] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. *Proceedings of the 16th international conference on World Wide Web*, 2007.

[13] H. Borko and M. Bernick. Automatic document classification. *Journal of the ACM (JACM)*, 10(2):151–162, 1963.

[14] B. Brainerd. Two Models for the Type-Token Relation with Time Dependant Vocabulary Reservoir. *Vocabulary Structure and Lexical Richness, Champion-Slatkine, Paris*, 1988.

[15] E. Brill. A simple rule-based part of speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, 16, 1992.

[16] E. Brill. *A corpus-based approach to language learning.* Graduate School of Arts and Sciences, University of Pennsylvania, 1993.

[17] E. Brill. Some advances in transformation-based part of speech tagging. *Proceedings of the twelfth national conference on Artificial Intelligence (vol. 1) table of contents*, pages 722–727, 1994.

[18] E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[19] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[20] Robert Bringhurst. *The Elements of Typographic Style.* Hartley & Marks, 2nd edition, December 2001.

[21] A. Broder. A taxonomy of web search. In *ACM Sigir Forum*, volume 36, page 10. ACM, 2002.

[22] J. Broughton. *Wikipedia: the missing manual.* O'Reilly, 2008.

[23] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[24] Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, March 2006.

[25] K. Burke. Four master tropes. *Kenyon Review*, pages 421–438, 1941.

[26] K. Burke. *A Grammar of Motives.* University of California Press, 1969.

[27] J.F. Burrows. Word-Patterns and Story-Shapes: The Statistical Analysis of Narrative Style. *Literary and Linguistic Computing*, 2(2):61–70, 1987.

[28] Gideon O. Burton. Silva Rhetoricae, 1996.

[29] H.E. Butler. The Institutio Oratoria of Quintilian, vol. 3. Loeb Classical Library, 1921.

[30] D. Cai and C. J. van Rijsbergen. Learning semantic relatedness from term discrimination information. *Expert Systems With Applications*, 36(2P1):1860–1875, 2009.

[31] R.P. Carver. Effect of a" chunked" typography on reading rate and comprehension. *Journal of Applied Psychology*, 54(3):288–296, 1970.

[32] D. Chandler. *Semiotics for beginners.* Daniel Chandler, 1994.

[33] D. Chandler. *Semiotics: the basics.* Routledge, 2002.

[34] P. Chesley, B. Vincent, L. Xu, and R.K. Srihari. Using verbs and adjectives to automatically classify blog sentiment. *Training*, 580(263):233.

[35] Adam Cheyer and David Martin. The open agent architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1):143–148, March 2001. NOTE: OAA.

[36] Cicero. Ad G. Herennium: De ratione dicendi (Rhetorica ad Herennium). *London-Cambridge (Mass.)*, 1954. With an English translation by Henry Caplan.

[37] R. Cilibrasi and P.M.B. Vitanyi. The google similarity distance. *Arxiv preprint cs/0412098*, 2004.

[38] Charles A. Clarke. Corpus of OCR-generated text from 19^th Century Canadian texts. Personal communication, 2010.

[39] Cochrane, Tom. Life is a Highway, 1991.

[40] Sarah Anne Curzon. *Laura Secord the Heroine of 1812*. BiblioBazaar, LLC, 2007.

[41] E. Dale and J.S. Chall. The concept of readability. *Elementary English*, 26(1):19–26, 1949.

[42] R. Dale. Cooking up referring expressions. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, page 75. Association for Computational Linguistics, 1989.

[43] Hercules Dalianis. Aggregation in natural language generation. *Computational Intelligence*, 15(4):384–414, 11/30 1999. M3: doi:10.1111/0824-7935.00099.

[44] Sandipan Dandapat, Sudeshna Sarkar, and Anupam Basu. A hybrid model for part-of-speech tagging and its application to bengali. In *ICCI 2004: International Conference on Computational Intelligence*, Hasretkent B3-Blk 9, Canakkale, 17020, Turkey, 17-19 Dec. 2004. Indian Institute of Technology Kharagpur, West Bengal, India, World Scientific and Technological Research Society.

[45] Thomas D'Arcy McGee and Mrs. J. Sadlier. *The poems of Thomas D'Arcy McGee: With copious notes. Also an introd. and biographical sketch.* D. & J. Sadlier, 1870.

[46] F. De Saussure. Course in General Linguistics, trans. *R. Harris, London: Duckworth*, 1983.

[47] F. De Saussure. *Course in General Linguistics.* Open Court, 1986.

[48] J. Diederich, J. Kindermann, E. Leopold, and G. Paass. Authorship Attribution with Support Vector Machines. *Applied Intelligence*, 19(1):109–123, 2003.

[49] C. DiMarco, G. Hirst, and E. Hovy. Generation by selection and repair as a method for adapting text for the individual reader. In *Proceedings of the Workshop on Flexible Hypertext, 8th ACM International Hypertext Conference*, 1997.

[50] C.C. Du Marsais, J. Paulhan, and C. Mouchard. *Traité des tropes.* Le Nouveau Commerce, 1977.

[51] William H. DuBay. The principles of readability, 2004.

[52] William H. DuBay. Robert gunning's fog readability formula. *Plain Language At Work Newsletter*, 23 March 2004.

[53] T. Dunning. Statistical identification of language. *Computing Research Laboratory Technical Memo MCCS*, pages 94–273, 1994.

[54] SR Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

[55] Jeanne Fahnestock. *Rhetorical Figures in Science*. Oxford University Press, New York, 1999.

[56] Jeanne Fahnestock. Private consultation on properties of rhetorical anaphora. Personal communication, 2010.

[57] J.W. Fan and C. Friedman. Semantic classification of biomedical concepts using distributional similarity. *Journal of the American Medical Informatics Association*, 14(4):467–477, 2007.

[58] C. Fellbaum. *WordNet: An electronic lexical database*. MIT press, 1998.

[59] Rudolf Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

[60] P. Fontanier. *Les figures du discours*. Paris, Flammarion, 1968.

[61] A.A. Fraenkel and Y. Bar-Hillel. *Foundations of set theory*. North-Holland pub. co., 1958.

[62] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 6–12, 2007.

[63] S.R. Garner. Weka: The waikato environment for knowledge analysis. In *Proceedings of the New Zealand Computer Science Research Students Conference*, pages 57–64. Citeseer, 1995.

[64] Jakub Gawryjolek. Automated annotation and visualization of rhetorical figures. Master's thesis, University of Waterloo, 2009.

[65] D. Gillick. Sentence Boundary Detection and the Problem with the US. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244. Association for Computational Linguistics, 2009.

[66] Neil M. Goldman and Neil M. Goldman. The boundaries of language generation. In *TINLAP '75: Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 74–78, Morristown, NJ, USA, 1975. Association for Computational Linguistics.

[67] B.J. Grosz, S. Weinstein, and A.K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225, 1995.

[68] Robert Gunning. The technique of clear writing, 1952.

[69] G. Hargis, M. Carey, A.K. Hernandez, P. Hughes, D. Longo, S. Rouiller, and E. Wilde. *Developing Quality Technical Information: A Handbook for Writers and Editors.* Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[70] Randy A. Harris. Private consultation on properties of rhetorical anaphora. Personal communication, 2010.

[71] Raquel Hervás and Pablo Gervás. Agent-based solutions for natural language generation tasks. *Lecture Notes in Computer Science*, 4177:103, 2006.

[72] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, pages 305–332, 1998.

[73] A. Hliaoutakis, G. Varelas, E. Voutsakis, and EGM Petrakis. Information retrieval by semantic similarity. *International Journal on Semantic Web & Information Systems*, 2(3):55–73, 2006.

[74] D.I. Holmes. Authorship attribution. *Computers and the Humanities*, 28(2):87–106, 1994.

[75] D.I. Holmes, M. Robertson, and R. Paez. Stephen Crane and the New-York Tribune: A Case Study in Traditional and Non-Traditional Authorship Attribution. *Computers and the Humanities*, 35(3):315–331, 2001.

[76] G. Holmes, A. Donkin, and I.H. Witten. Weka: A machine learning workbench. In *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems*, pages 357–361. Citeseer, 1994.

[77] P.C. Jackson. *Introduction to artificial intelligence.* Dover publications, 1985.

[78] B.J. Jansen, D.L. Booth, and A. Spink. Determining the informational, navigational, and transactional intent of Web queries. *Information Processing and Management*, 44(3):1251–1266, 2008.

[79] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *Arxiv preprint cmp-lg/9709008*, 1997.

[80] H. Joon, H. Myoung, and Lee Yoon Joon. Information retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation*, 49, 1993.

[81] Benjamin Jowett. *Plato: Gorgias.* Echo Library, 1999.

[82] P. Juola and R.H. Baayen. A Controlled-corpus Experiment in Authorship Identification by Cross-entropy, 2005.

[83] I.H. Kang and G.C. Kim. Query type classification for web document retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, page 71. ACM, 2003.

[84] H. Kautz and J.F. Allen. Generalized plan recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 32–37, 1986.

[85] G.A. Kennedy. *On rhetoric: A theory of civic discourse.* Oxford University Press, USA, 1991.

[86] T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.

[87] M. Koppel and J. Schler. Authorship verification as a one-class classification problem. *ACM International Conference Proceeding Series*, 2004.

[88] OV Kukushkina, AA Polikarpov, and DV Khmelev. Using Literal and Grammatical Statistics for Authorship Attribution. *Problems of Information Transmission*, 37(2):172–184, 2001.

[89] C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[90] YH Li and AK Jain. Classification of text documents. *The Computer Journal*, 41(8):537, 1998.

[91] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, 1998.

[92] D.J. Litman and J.F. Allen. A plan recognition model for subdialogues in conversations**. *Cognitive Science*, 11(2):163–200, 1987.

[93] I. MacKinnon. Wikipedia-Based Semantic Enhancements for Information Nugget Retrieval. Master's thesis, University of Waterloo, 2008. Master of Mathematics in Computer Science.

[94] I. Mani. Recent developments in text summarization. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 529–531. ACM New York, NY, USA, 2001.

[95] C.D. Manning and H. Schütze. Foundations of Statistical Natural Language Processing. *Computational Linguistics*, 26(2).

[96] D. Marcu. *The theory and practice of discourse parsing and summarization.* MIT Press, 2000.

[97] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. Citeseer.

[98] Michael C. McCord. Review of discourse production: a computer model of some aspects of a speaker by anthony davey. edinburgh univ. press 1978. *Comput.Linguist.*, 6(2):112–112, 1980. note: Reviewer-Michael C. McCord.

[99] Kathleen R. McKeown. *Text generation: using discourse strategies and focus constraints to generate natural language text.* Cambridge University Press, New York, NY, USA, 1985.

[100] G. McKoon and R. Ratcliff. Inference during reading. *Psychological review*, 99(3):440–466, 1992.

[101] A. Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318, 2002.

[102] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. Introduction to wordnet: An on-line lexical database*. *International Journal of lexicography*, 3(4):235–244, 1990.

[103] G.A. Miller et al. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81–97, 1956.

[104] George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11), 1995.

[105] G. Mishne. Experiments with mood classification in blog posts. In *Proceedings of ACM SIGIR 2005 Workshop on Stylistic Analysis of Text for Information Access*. Citeseer, 2005.

[106] P.M. Mitchell, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[107] S. Mohammad and G. Hirst. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, 2006.

[108] Paul W. Oman and Curtis R. Cook. Typographic style is more than cosmetic. *Commun. ACM*, 33(5):506–520, 1990.

[109] I. Ounis, M. De Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the trec-2006 blog track. In *Proceedings of TREC*, volume 6. Citeseer, 2006.

[110] D.D. Palmer and M.A. Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–267, 1997.

[111] H. Peacham. The Garden of Eloquence (1593), ed. *William G. Crane (Gainesville FL: Scholars' Facsimiles and Reprints, 1954)*, page 4, 1954.

[112] F.C. Pereira, R. Hervás, P. Gervás, and A. Cardoso. A multiagent text generator with simple rhetorical habilities. In *Computational Aesthetics: Artificial Intelligence Approaches to Beauty and Happiness (Workshop from AAAI-06)*, pages 37–44, Boston, USA, 07/2006 2006.

[113] G. Pirró and N. Seco. Design, Implementation and Evaluation of a New Semantic Similarity Metric Combining Features and Intrinsic Information Content. *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems*, pages 1271–1288, 2008.

[114] R. Puttenham. *George Puttenham: The Arte of English Poesie.* A. Murray & son, 1869.

[115] L. Rabiner and B. Juang. An introduction to hidden Markov models. *ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine]*, 3(1 Part 1):4–16, 1986.

[116] LR Rabiner. A tutorial on hidden Markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[117] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE transactions on systems, man and cybernetics*, 19(1):17–30, 1989.

[118] M. Reape and C. Mellish. Just what is aggregation anyway. Citeseer.

[119] T. Reinhardt and M. Winterbottom. *Quintilian: Institutio Oratoria.* Oxford University Press, 2006.

[120] T. Reinhardt and M. Winterbottom. *Quintilian: Institutio Oratoria, Book 2.* Oxford University Press, 2006.

[121] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems.* Cambridge University Press, 2000.

[122] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *Arxiv preprint cmp-lg/9511007*, 1995.

[123] J.C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19, 1997.

[124] W.R. Roberts. *Greek Rhetoric and Literary Criticism: By W. Rhys Roberts...* Longmans, Green and co., 1928.

[125] M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets. *Advances in Web Intelligence*, 3528:380–386, 2005.

[126] A.D. Scriver. *Semantic Distance in WordNet a Simplified and Improved Measure of Semantic Relatedness.* Library and Archives Canada= Bibliothèque et Archives Canada, 2006.

[127] N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in WordNet. In *ECAI*, volume 16, page 1089, 2004.

[128] N. Shimizu, M. Hagiwara, Y. Ogawa, K. Toyama, and H. Nakagawa. Metric Learning for Synonym Acquisition. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008.

[129] K. Silverman. *The Subject of Semiotics.* Oxford University Press, USA, 1983.

[130] H.H. Sommers. Statistical Methods in Literary Analysis. *The Computer & Literary Style: Introductory Essays and Studies*, 1966.

[131] K. Spärck Jones. Automatic summarising: The state of the art. *Information Processing and Management*, 43(6):1449–1481, 2007.

[132] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic authorship attribution. *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 158–164, 1999.

[133] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Computer-Based Authorship Attribution Without Lexical Measures. *Computers and the Humanities*, 35(2):193–214, 2001.

[134] Dan M. Stevens. 101 standards for online communication, 1997.

[135] M. Strube. Never look back: An alternative to centering. In *Annual Meeting – Association for Computational Linguistics*, volume 36, pages 1251–1257. Association for Computational Linguistics, 1998.

[136] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management*, pages 67–74. ACM New York, NY, USA, 1993.

[137] G. Takeuti and W.M. Zaring. *Introduction to axiomatic set theory.* Springer-Verlag, 1982.

[138] W. Taylor. *Tudor figures of rhetoric.* Language Pr, 1972.

[139] J.R. Tetreault. A corpus-based evaluation of centering and pronoun resolution. *Computational Linguistics*, 27(4):507–520, 2001.

[140] M. Thimm and G. Kern-Isberner. A Distributed Argumentation Framework using Defeasible Logic Programming (Extended Version). Technical report, Technical report, Technische Universitat Dortmund, 2008.

[141] M. Thimm and G. Kern-Isberner. On the Relationship of Defeasible Argumentation and Answer Set Programming. In *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA08)*, volume 172, pages 393–404, 2008.

[142] Arthur T. Turnbull and Russell N. Baird. *The Graphics of Communication: Typography–Layout–Design. Third Edition.* Holt, Rinehart and Winston, Inc., 383 Madison Avenue, New York, New York 10017 ($11.95 cloth), 1975.

[143] A. Tversky et al. Features of similarity. *Psychological review*, 84(4):327–352, 1977.

[144] Vessela Valiavitcharska. Private consultation on properties of rhetorical anaphora. Personal communication, 2010.

[145] W. Vonk. The use of referential expressions in structuring discourse. *Discourse Representation And Text Processing: Special Double Issue Of Language And Cognitive Processes*, 7(3/4):301–333, 1993.

[146] L. Wanner and E. Hovy. The healthdoc sentence planner. In *INLG'96*, pages 1–10, Herstmonceux Castle, Sussex, 1996.

[147] T. Wilson. *The Arte of Rhetorique, 1553: A Facsimile Reporduction with an Introd. by Robert Hood Bowers*. Scholars'Facsimiles & Reprints, 1962.

[148] I.H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S.J. Cunningham. Weka: Practical machine learning tools and techniques with Java implementations. In *ICONIP/ANZIIS/ANNES*, volume 99, pages 192–196. Citeseer, 1999.

[149] D. Yang and D.M.W. Powers. Measuring semantic similarity in the taxonomy of WordNet. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 315–322. Australian Computer Society, Inc. Darlinghurst, Australia, Australia, 2005.

[150] Y. Zhao and J. Zobel. Effective and Scalable Authorship Attribution Using Function Words. *Lecture Notes in Computer Science*, 3689:174–189, 2005.

[151] Y. Zhao, J. Zobel, and P. Vines. Using relative entropy for authorship attribu-

tion. *Proc. 3rd AIRS Asian Information Retrieval Symposium, Springer*, pages 92–105, 2006.

Appendix A

Annotation Statistics

This section contains the primary statistics that resulted from the annotation procedure in Section 5.1.1. All values on the y-axis were normalised to indicate the representative percentage of each dataset. For example, of all the positive matches in Figure A.1 around 60 percent had a tuple width of two, 16 percent a tuple width of three, and so on. A different way to look at this is to take the values as being those of a representative group of 100 epanaphora from each set. The reason for choosing to display this data in this format – as opposed to raw numbers – is because during the annotation process the overall ratio of epanaphora having a positive match to the search criteria given to the annotators versus epanaphora not matching all criteria is quite low. If raw numbers had been used, it would not have been practical to compare these side by side.

Furthermore, for attributes requiring average and median value analysis those values were rounded to the nearest integer. The reasons for this are two-fold: First of all the input set itself is restricted to integer values, and for the purpose of this analysis it it reasonable to generate output that can be compared with the input on the same scale, which in this case would be integers. For example it just does not make sense to be measuring sentence length in fractions of words. Secondly we wanted to restrict the size of the input to the classification algorithms in Section 5.1.2

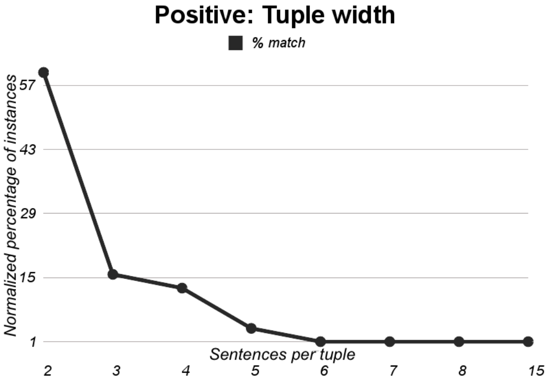


Figure A.1: Tuple width statistics for positive matches

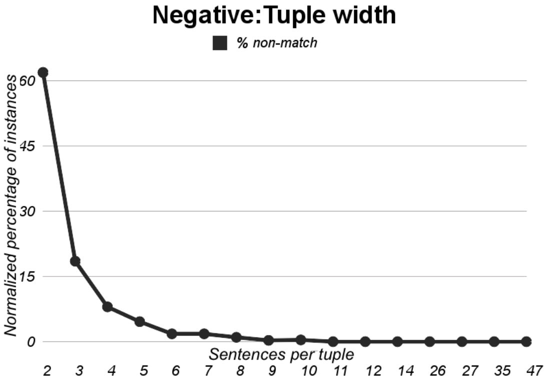


Figure A.2: Tuple width statistics for negative matches

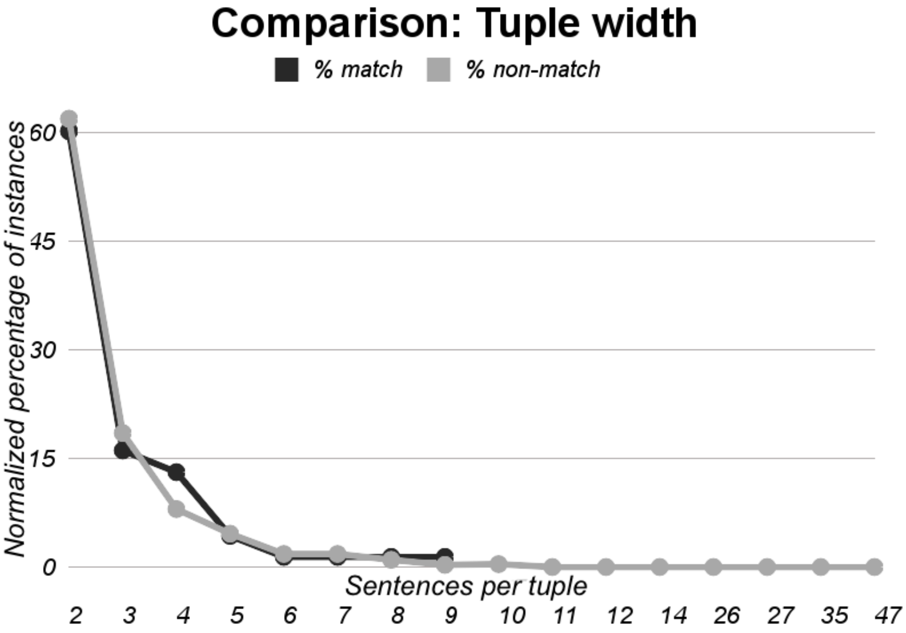


Figure A.3: Combined positive and negative tuple width statistics

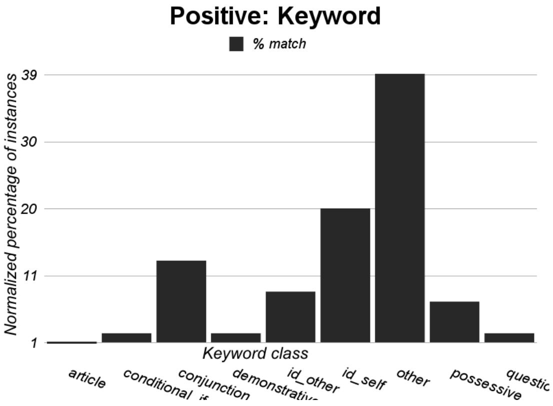


Figure A.4: Keyword statistics for positive matches

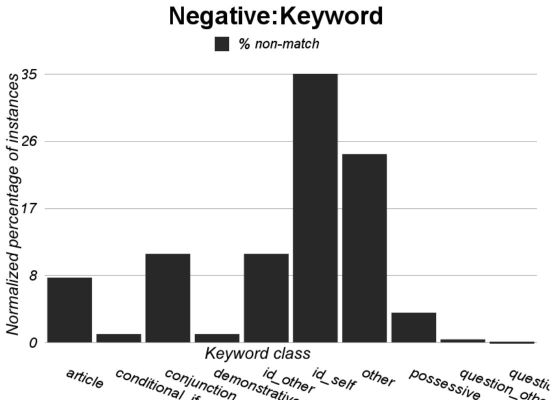


Figure A.5: Keyword statistics for negative matches

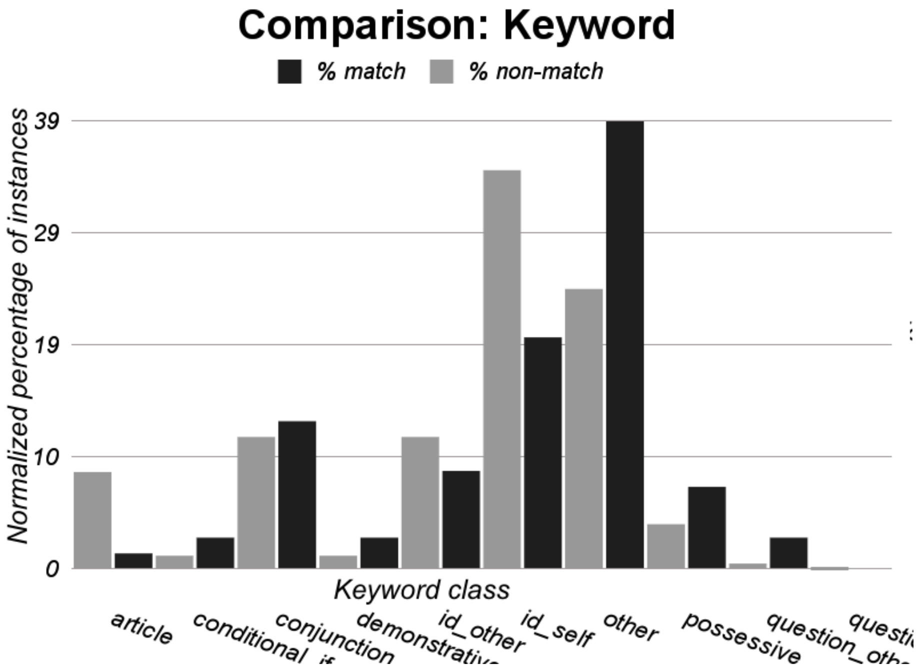


Figure A.6: Combined positive and negative keyword statistics

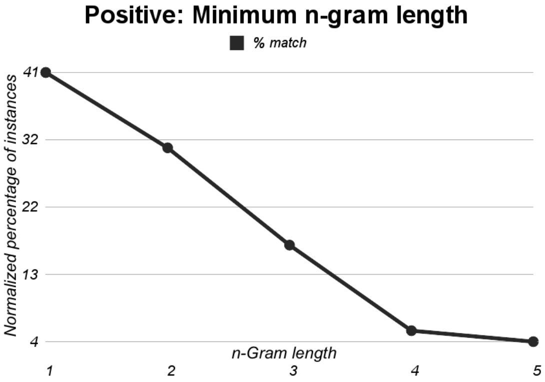


Figure A.7: n-Gram statistics for minimum positive matches

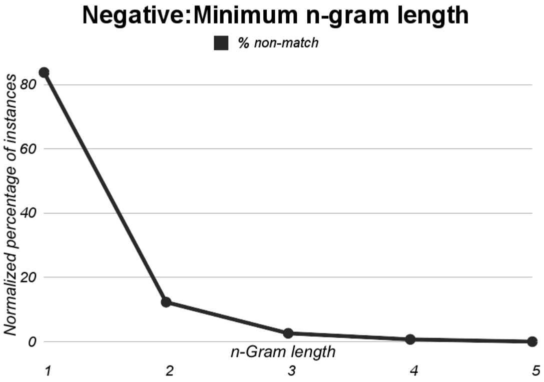


Figure A.8: n-Gram statistics for minimum negative matches

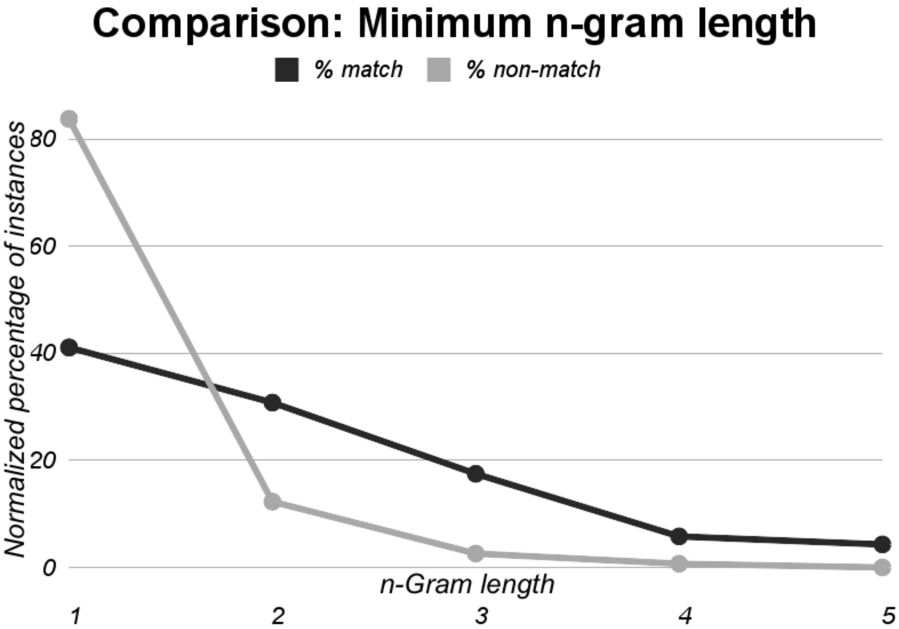


Figure A.9: Combined positive and negative minimum n-gram statistics

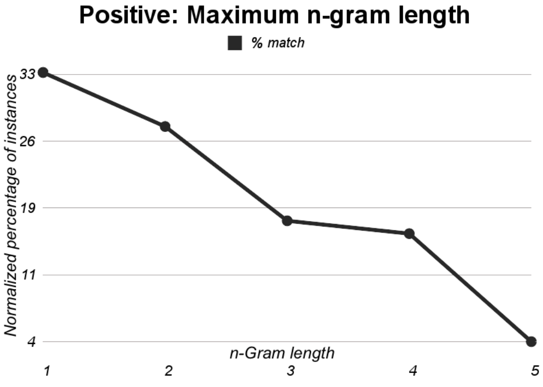


Figure A.10: n-Gram statistics for maximum positive matches

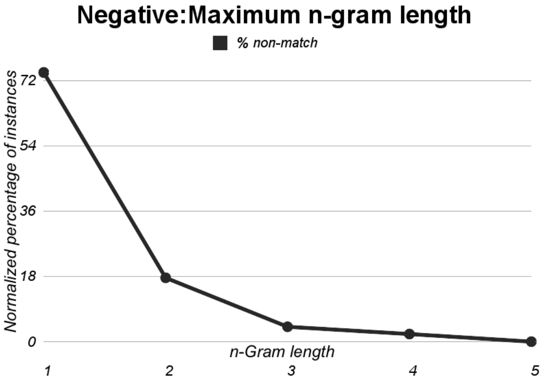


Figure A.11: n-Gram statistics for maximum negative matches

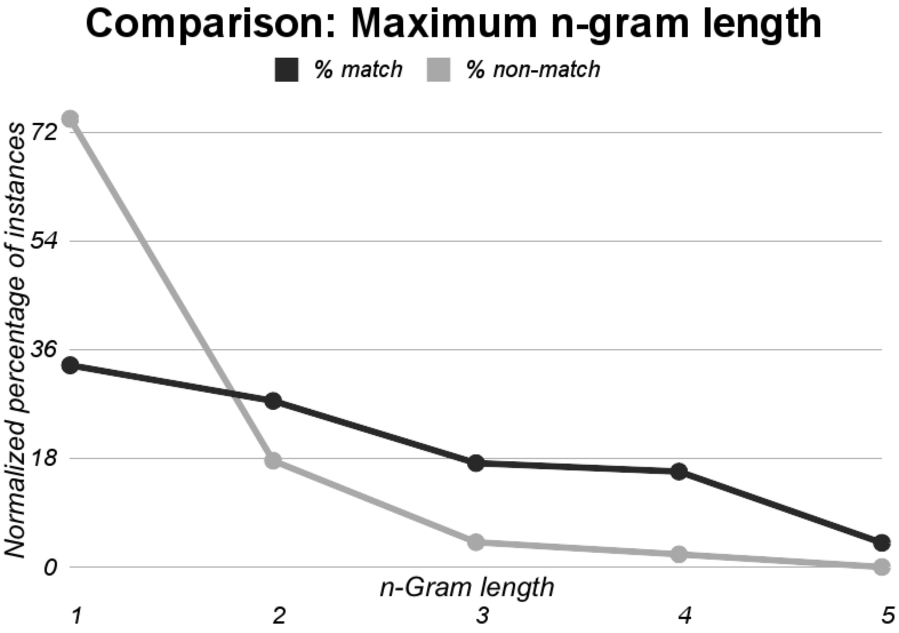


Figure A.12: Combined positive and negative maximum n-gram statistics

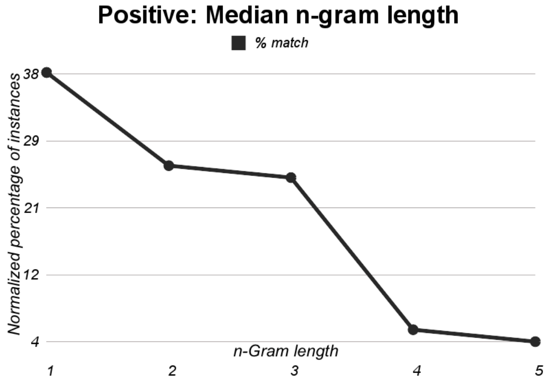


Figure A.13: n-Gram statistics for median positive matches

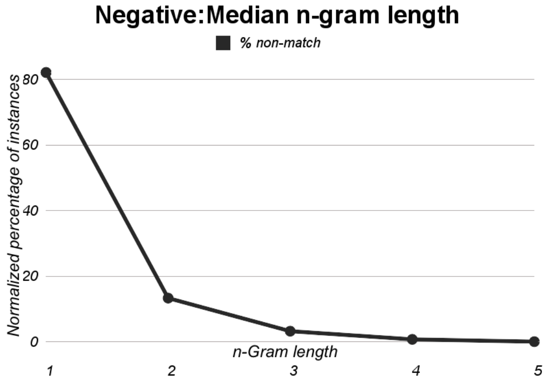


Figure A.14: n-Gram statistics for median negative matches

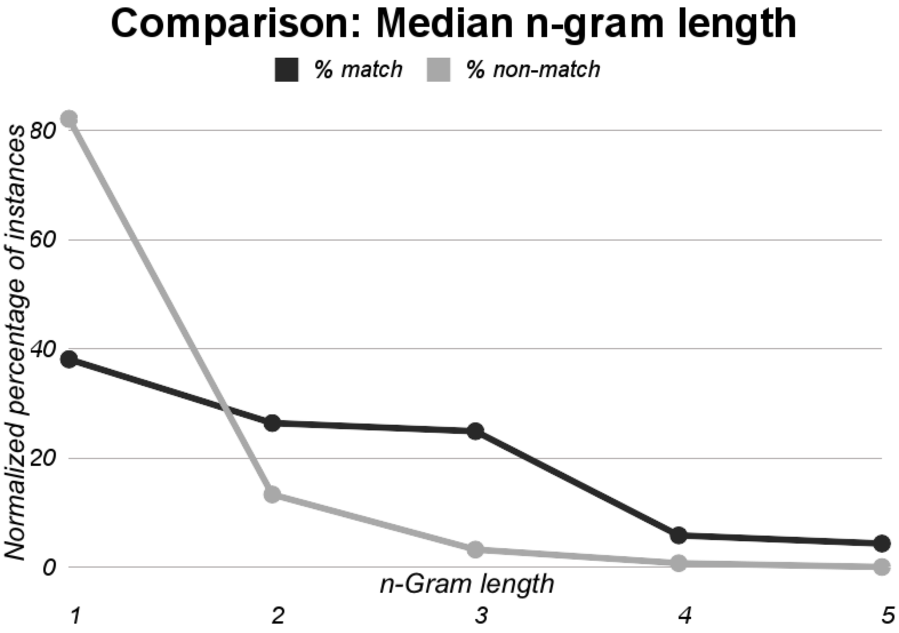


Figure A.15: Combined positive and negative median n-gram statistics

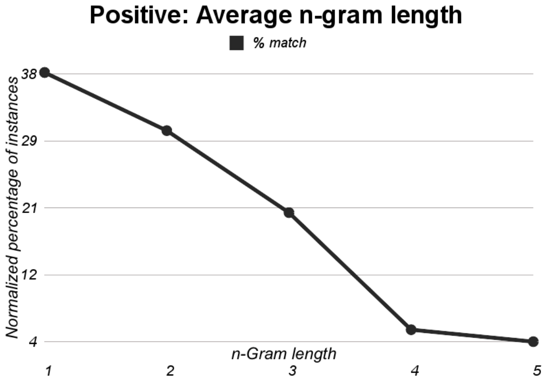


Figure A.16: n-Gram statistics for average positive matches

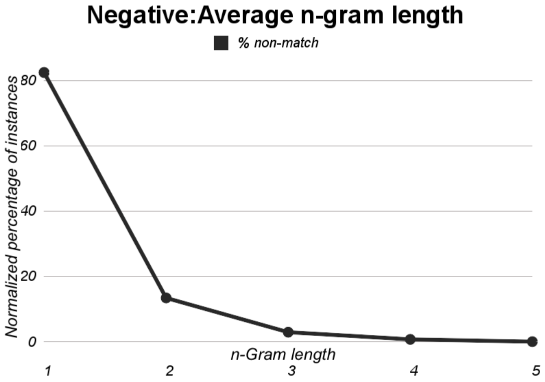


Figure A.17: n-Gram statistics for average negative matches

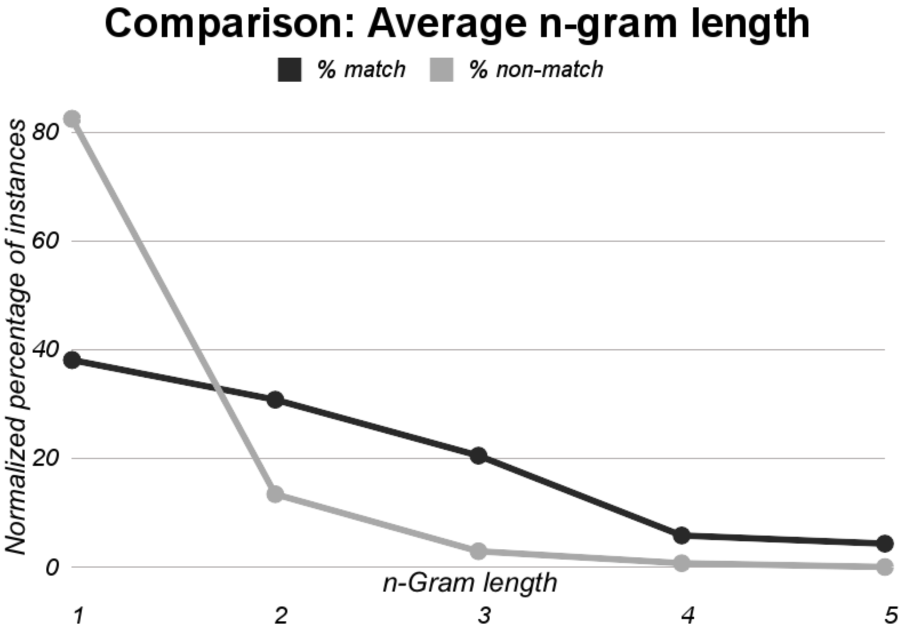


Figure A.18: Combined positive and negative average n-gram statistics

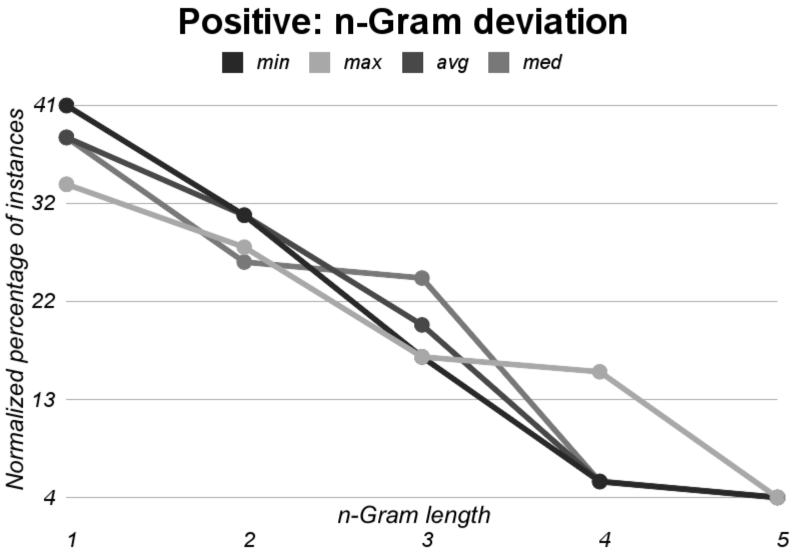


Figure A.19: Keyword statistics for positive matches

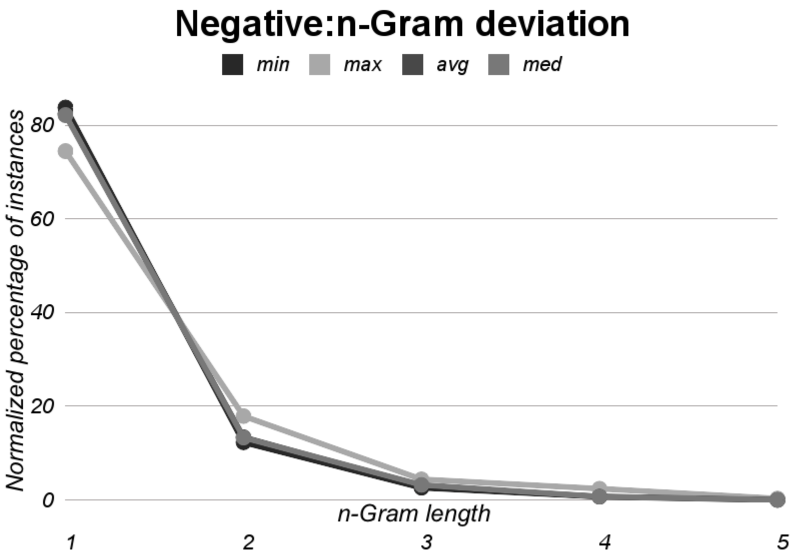
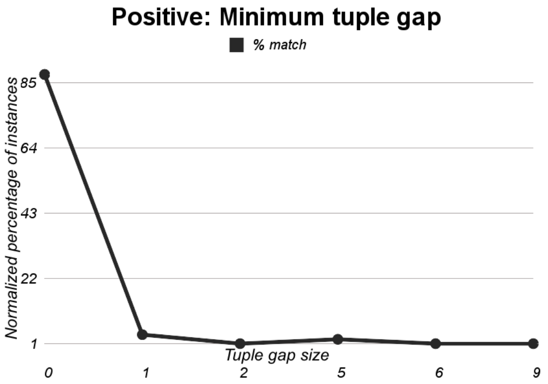


Figure A.20: Keyword statistics for negative matches



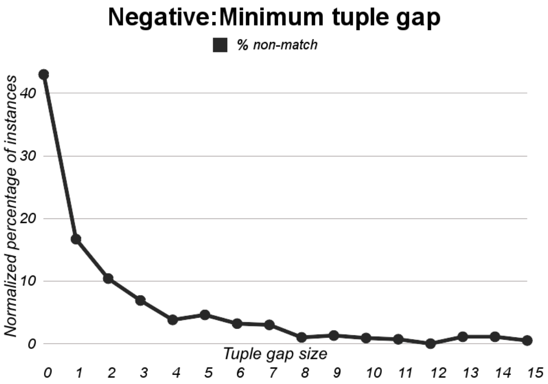


Figure A.21: Gap statistics for minimum positive matches

Figure A.22: Gap statistics for minimum negative matches

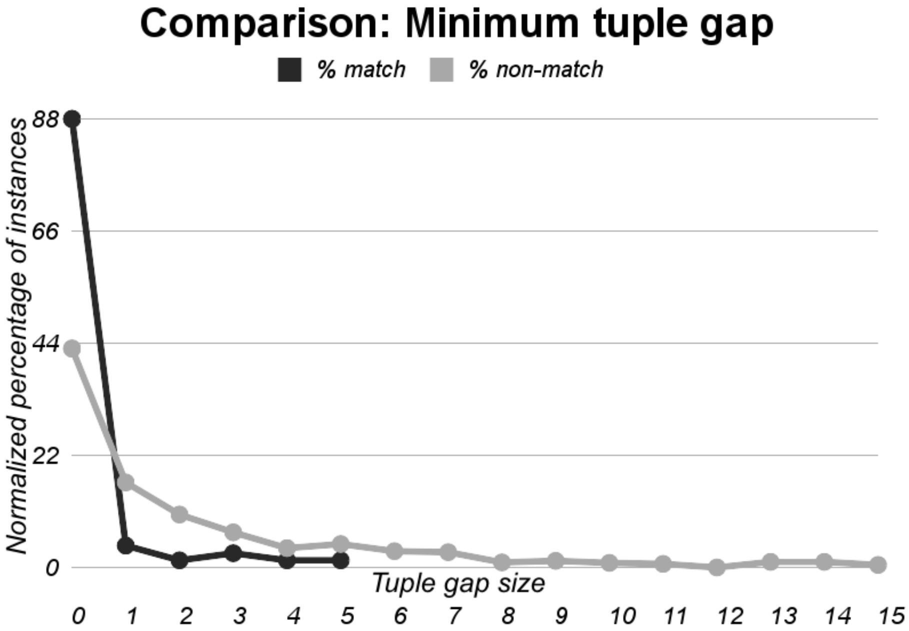


Figure A.23: Combined positive and negative minimum gap statistics

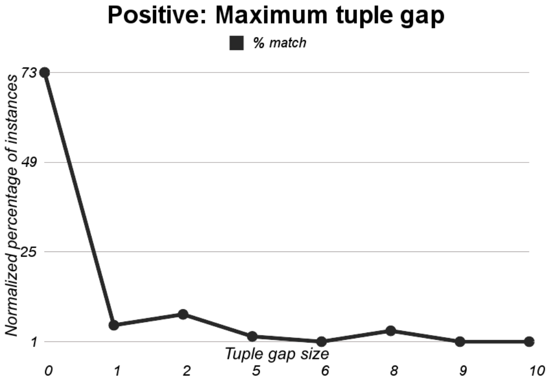


Figure A.24: Gap statistics for maximum positive matches

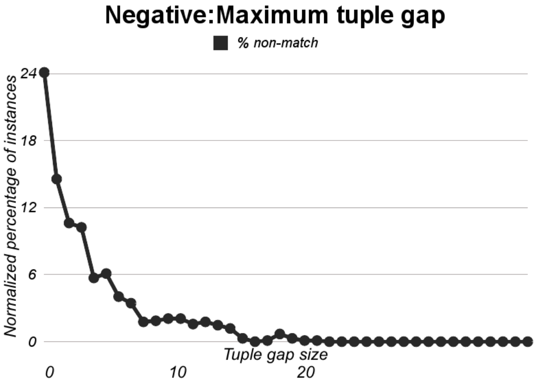


Figure A.25: Gap statistics for maximum negative matches

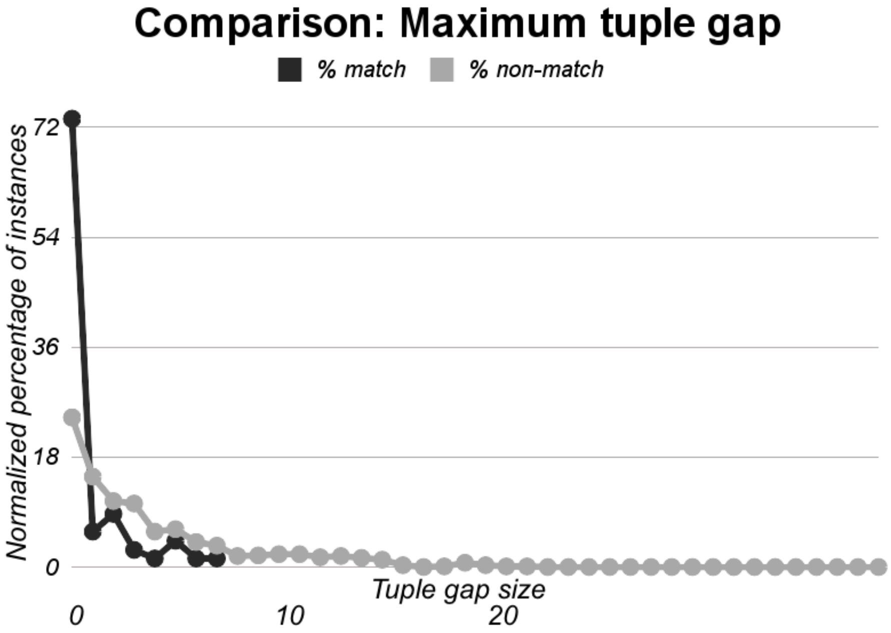
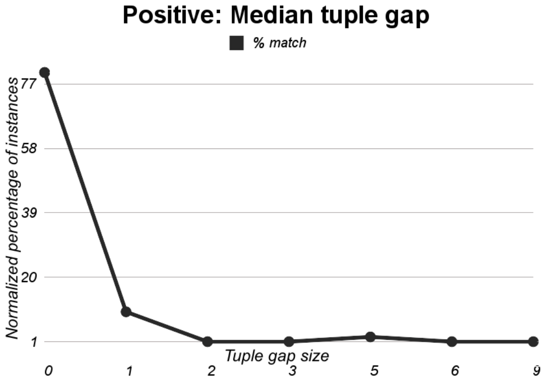


Figure A.26: Combined positive and negative maximum gap statistics



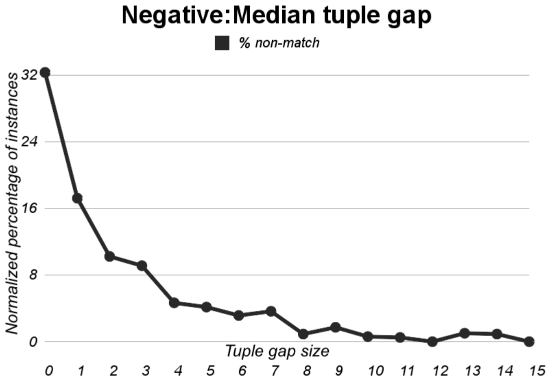


Figure A.27: Gap statistics for median positive matches

Figure A.28: Gap statistics for median negative matches

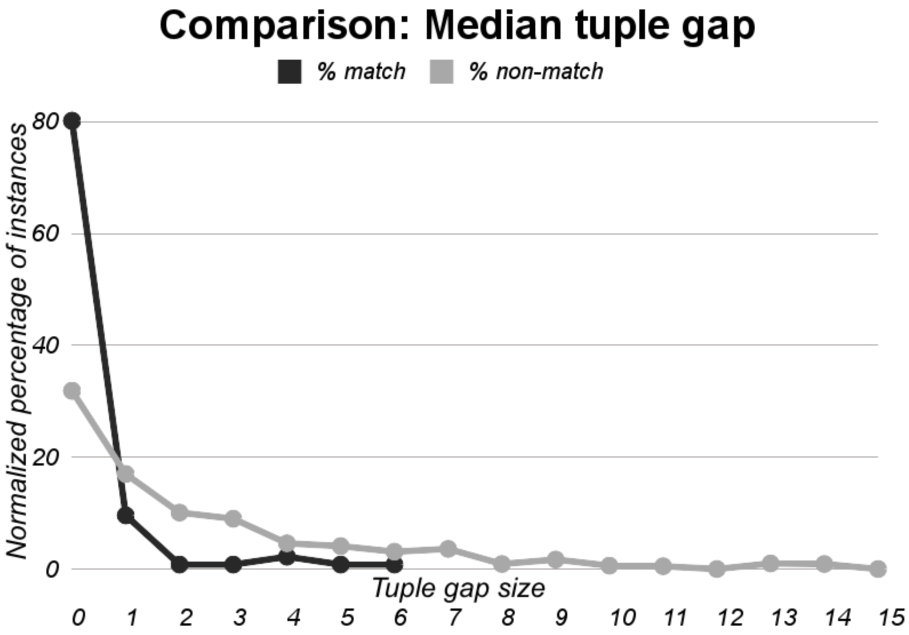


Figure A.29: Combined positive and negative median gap statistics

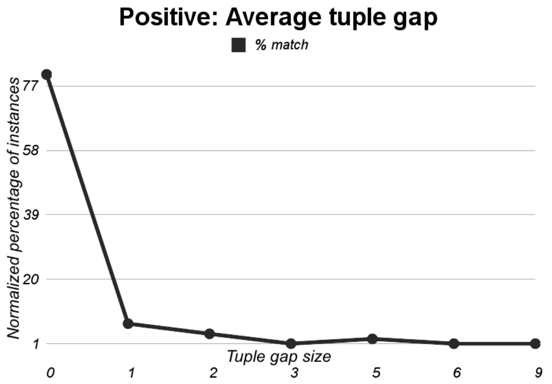


Figure A.30: Gap statistics for average positive matches

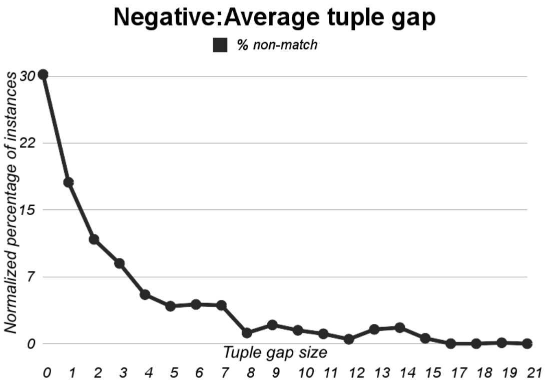


Figure A.31: Gap statistics for average negative matches

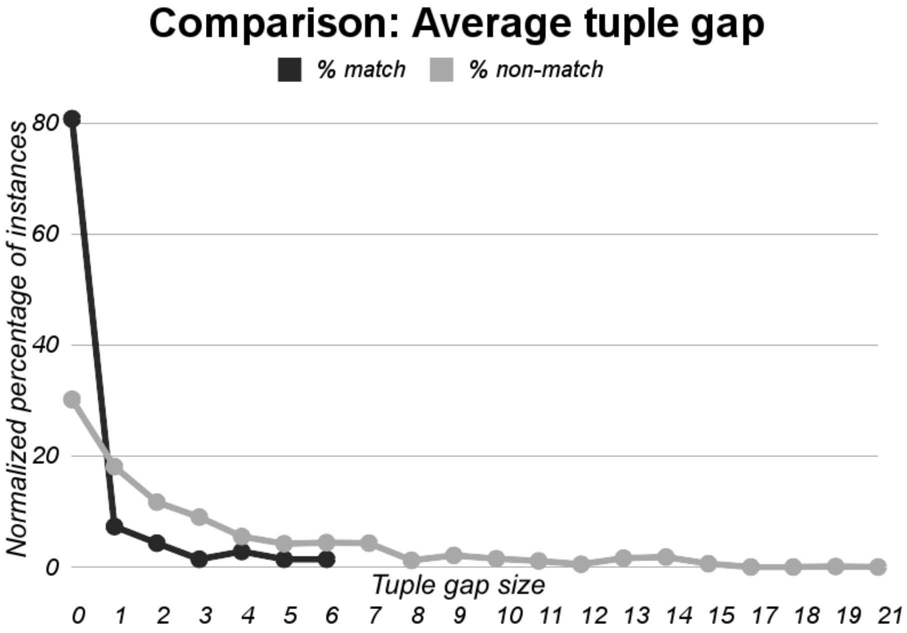


Figure A.32: Combined positive and negative average gap statistics

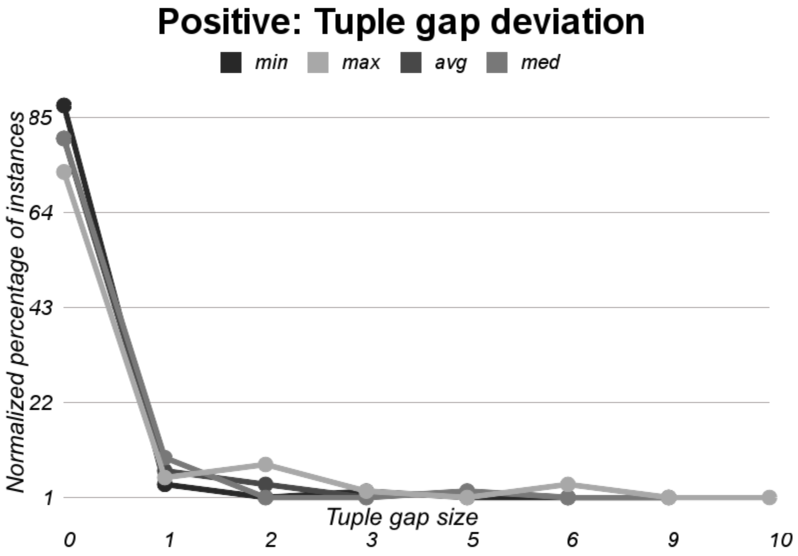


Figure A.33: Keyword statistics for positive matches

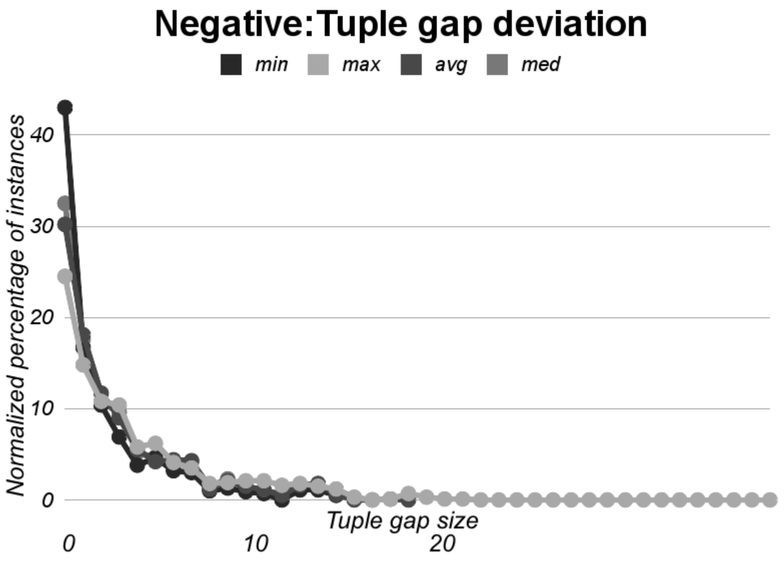


Figure A.34: Keyword statistics for negative matches

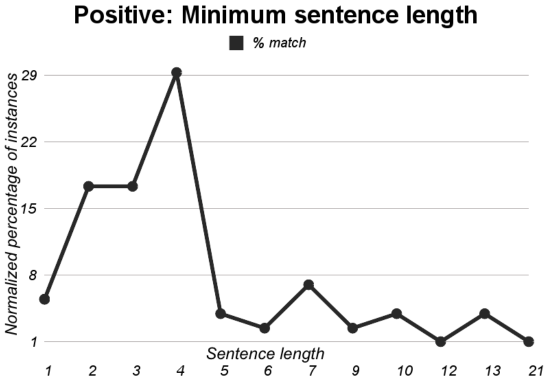


Figure A.35: Sentence length statistics for minimum positive matches

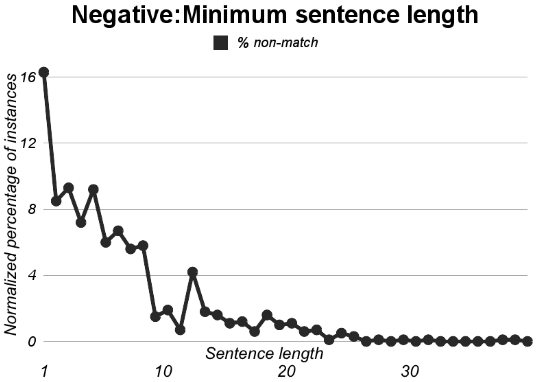


Figure A.36: Sentence length statistics for minimum negative matches

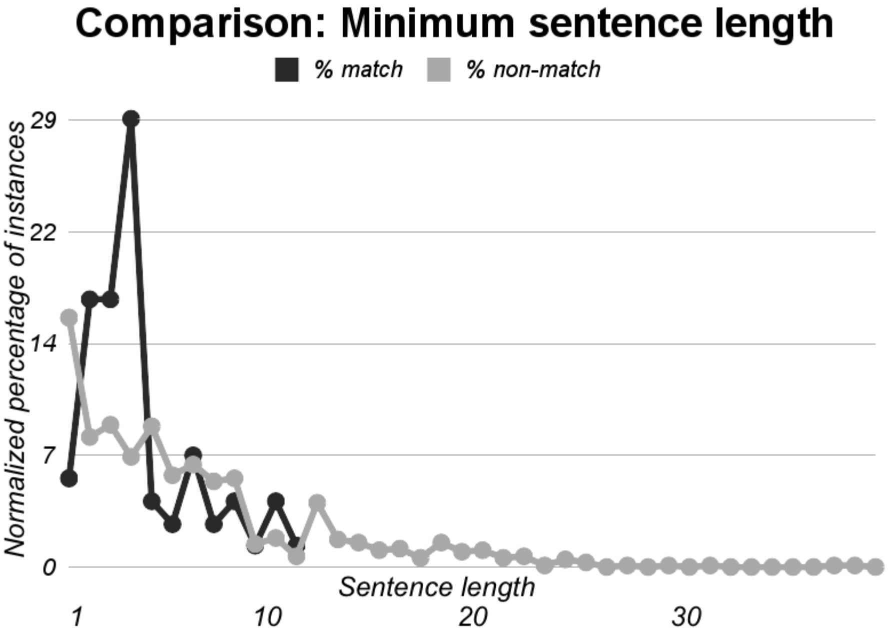


Figure A.37: Combined positive and negative minimum sentence length statistics

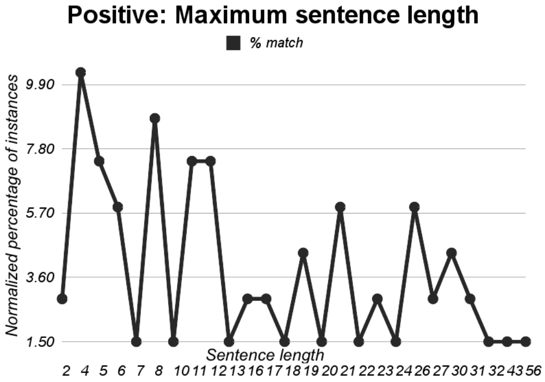


Figure A.38: Sentence length statistics for maximum positive matches

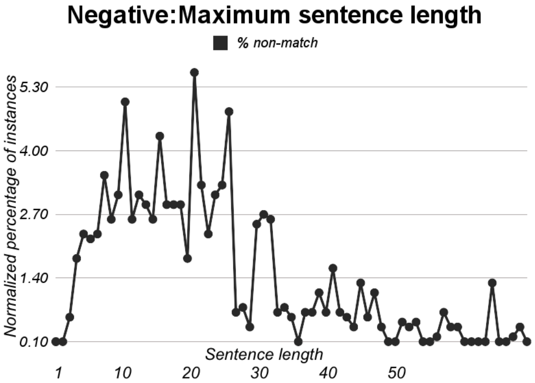


Figure A.39: Sentence length statistics for maximum negative matches

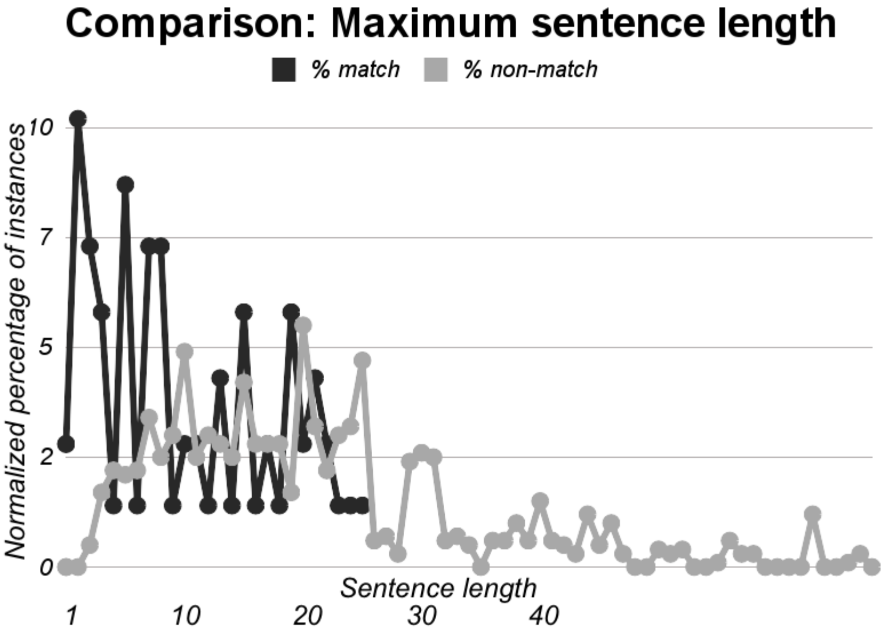


Figure A.40: Combined positive and negative maximum sentence length statistics

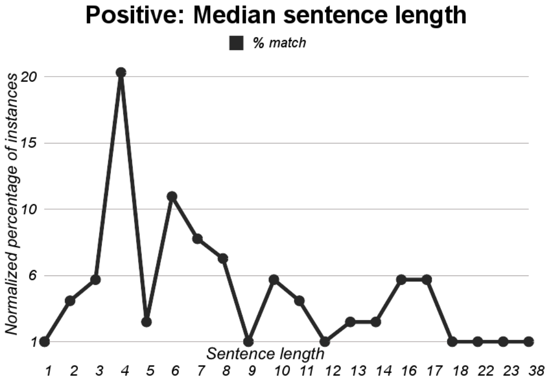


Figure A.41: Sentence length statistics for median positive matches

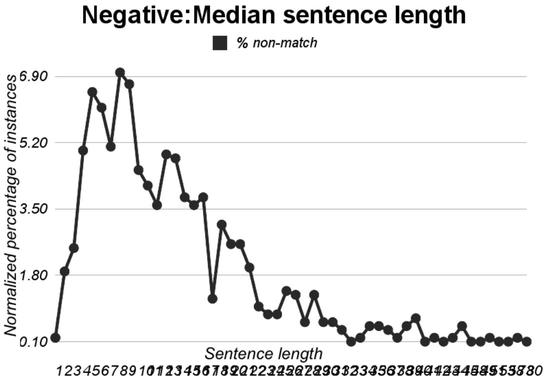


Figure A.42: Sentence length statistics for median negative matches

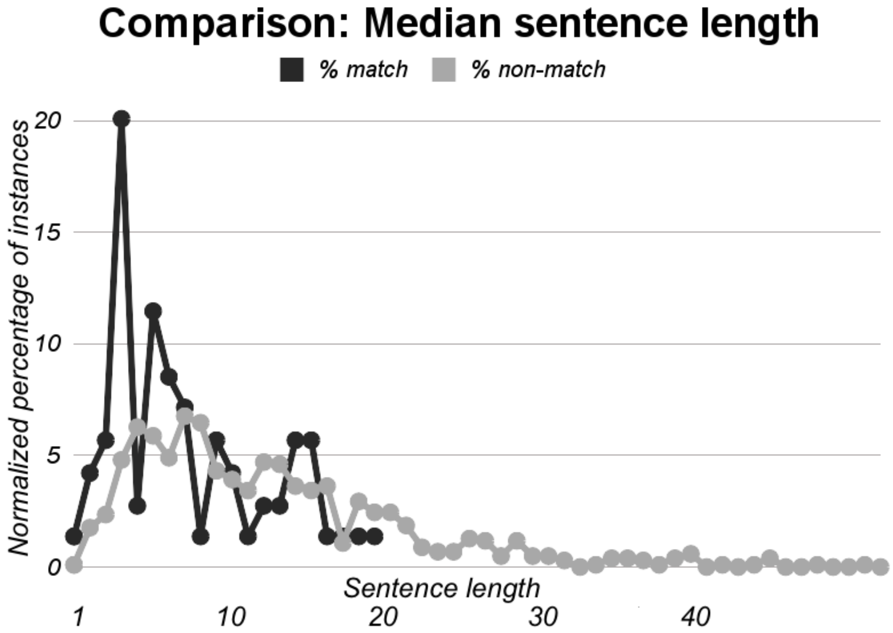


Figure A.43: Combined positive and negative median sentence length statistics

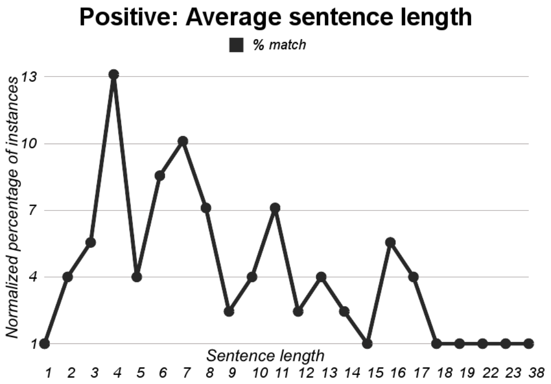


Figure A.44: Sentence length statistics for average positive matches

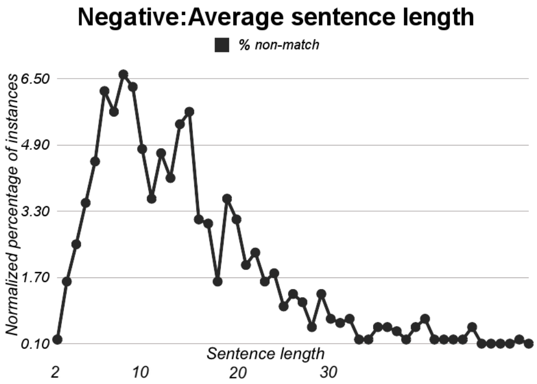


Figure A.45: Sentence length statistics for average negative matches

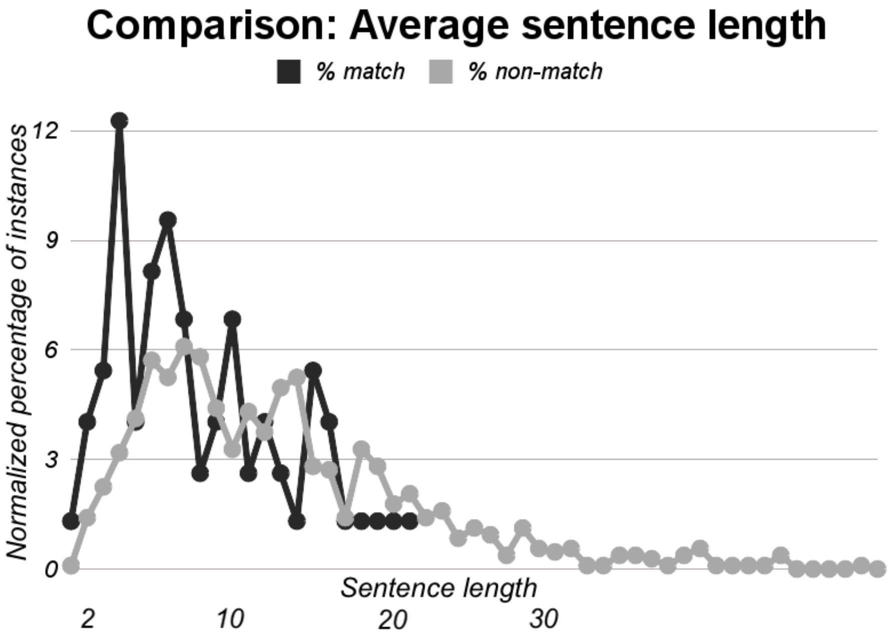


Figure A.46: Combined positive and negative average sentence length statistics

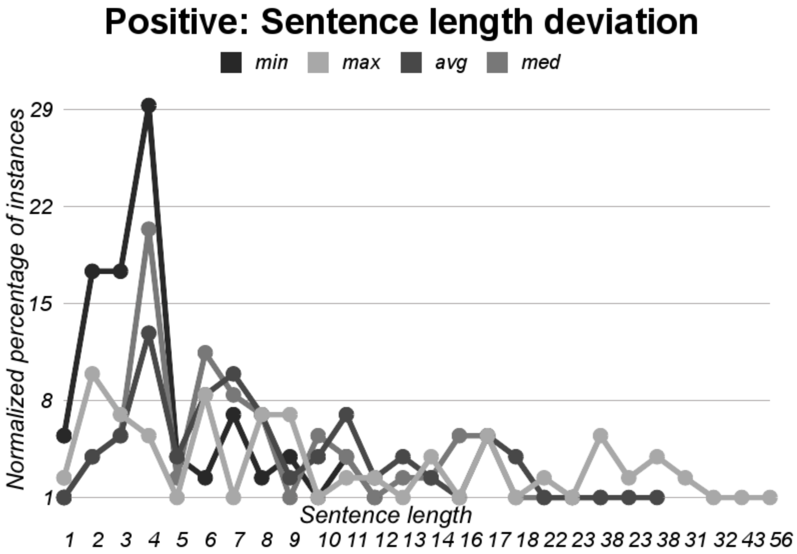


Figure A.47: Keyword statistics for positive matches

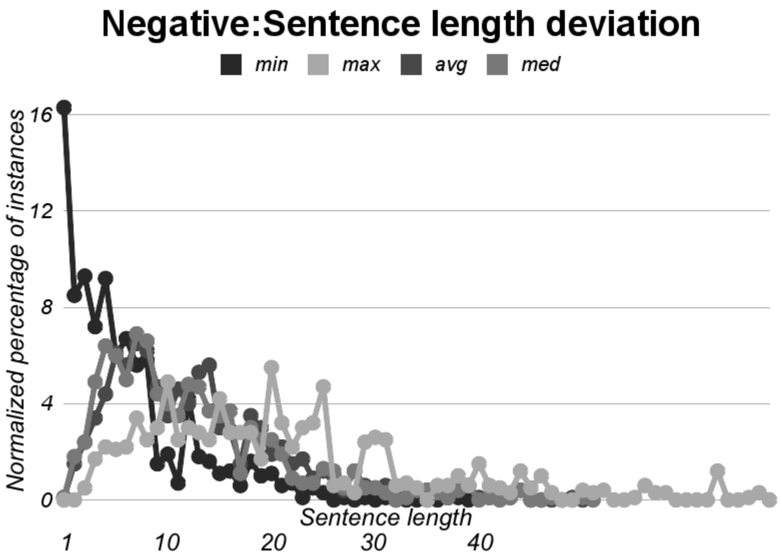


Figure A.48: Keyword statistics for negative matches

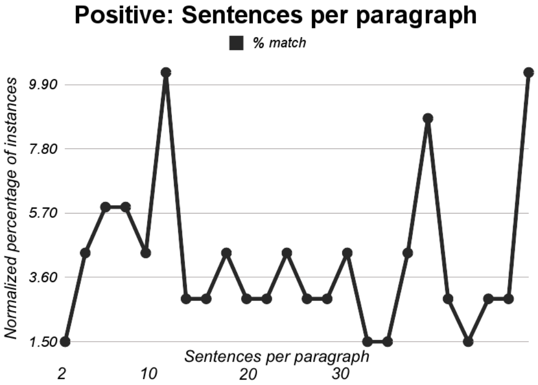


Figure A.49: Positive match statistics for sentences per paragraph

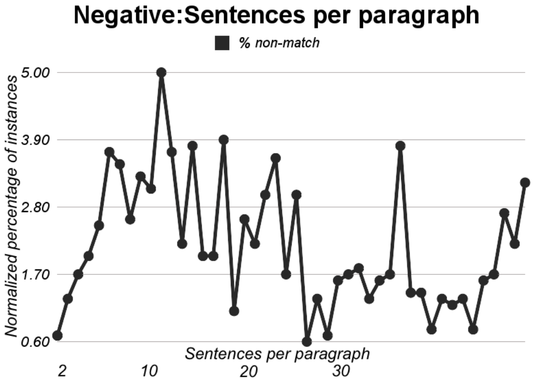


Figure A.50: Negative match statistics for sentences per paragraph

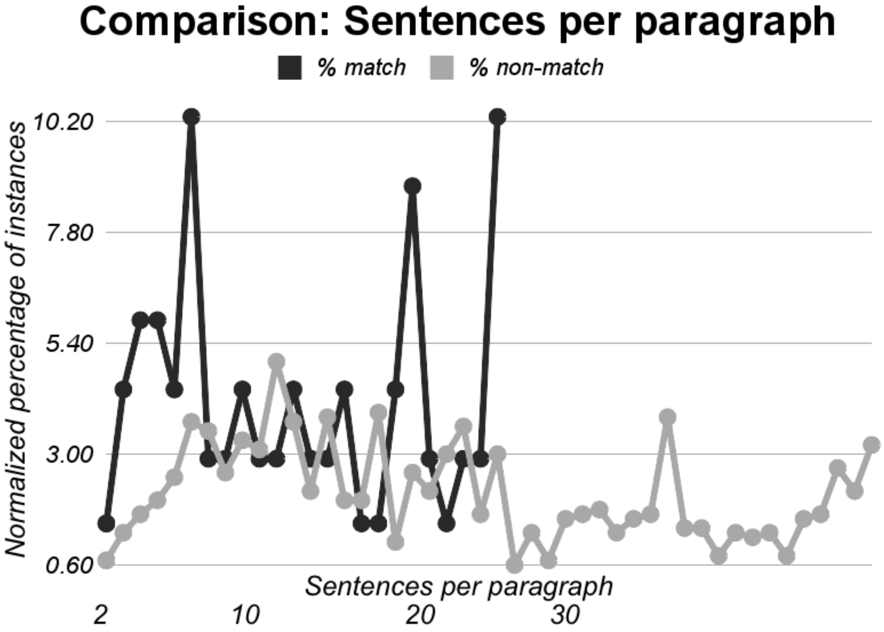


Figure A.51: Combined positive and negative statistics for sentences per paragraph

Appendix B

Classifier Results

.

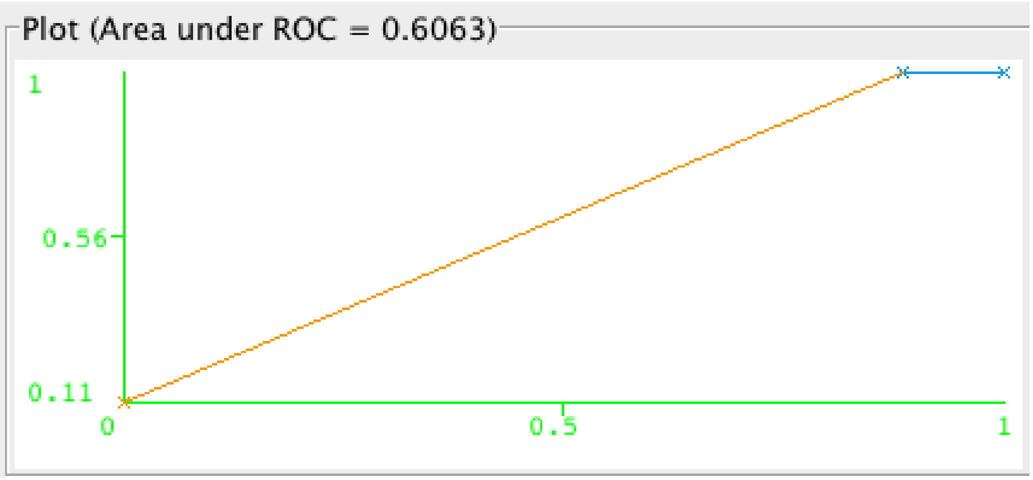


Figure B.1: J48 decision tree threshold (ROC) curve

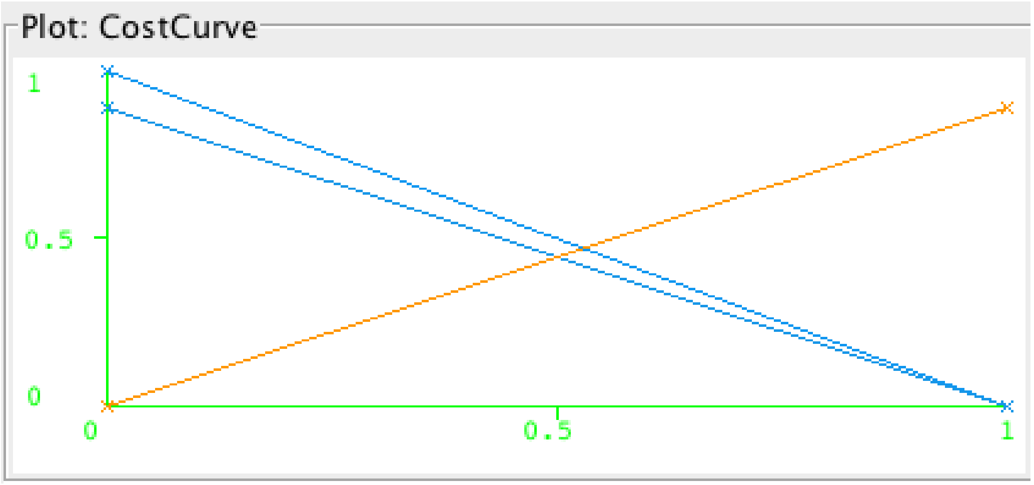


Figure B.2: J48 decision tree cost curve

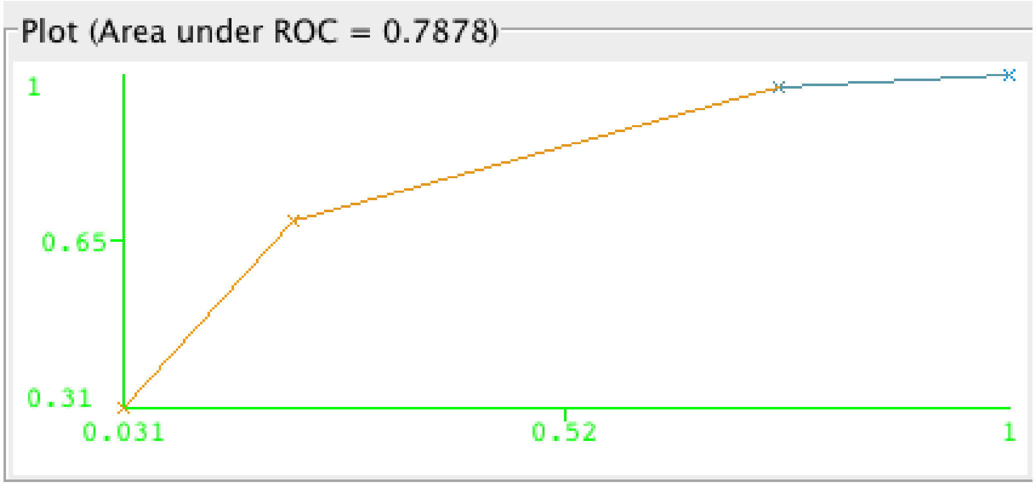


Figure B.3: J48 decision tree threshold (ROC) curve (revised training corpus)

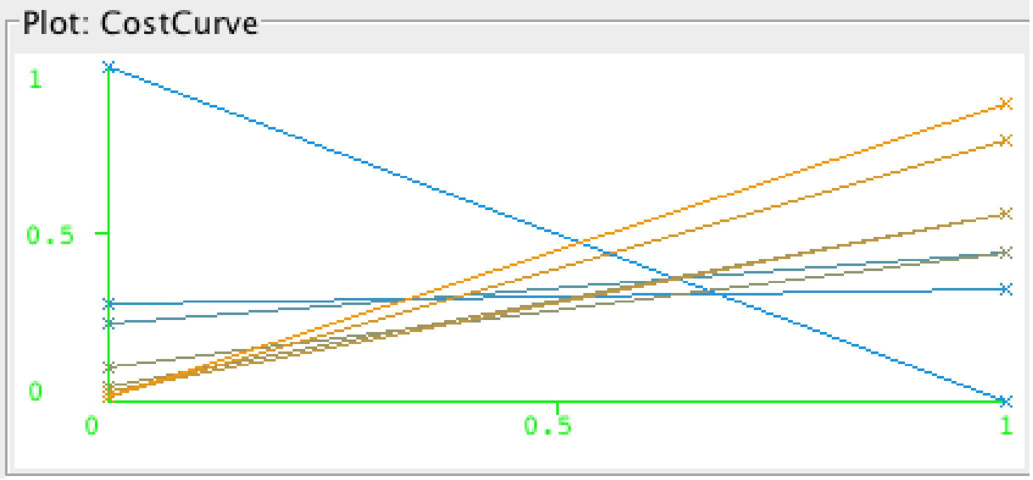


Figure B.4: J48 decision tree cost curve (revised training corpus)

Glossary

Aggregation In Computational Linguistics, the process by which sentences with similar structures are combined to reduce repetition, 12

Artificial Intelligence (AI) A branch of Computer Science that focuses on creating intelligent machines, 10, 184

Atomic Changes Small, local, and self-contained changes affecting only few tokens or entities, 21

Bayesian Network A probabilistic model that represents the relations between random variables in a Directed Acyclic Graph (DAG), 21

Computational Linguistics An area of AI which focuses on the understanding and generation of natural language (human readable) text, 184–186, 188

Coreference In Computational Linguistics, the process by which similar structures in consecutive sentences are replaced with referring expressions, such as pronouns, 12, 187

Directed Acyclic Graph (DAG) A directed graph with no cycles, 184

Distributed System A computer system that distributes the computational workload among several machines in a network, 11

Entropy A measure of the distribution of lexical tokens within a document, 21

Epanaphora Repetition of the same word or group of words at the beginning of successive clauses, sentences, or lines, 41

Epistrophe Ending a series of lines, phrases, clauses, or sentences with the same word or words. Ex. "What lies behind *us* and what lies before *us* are tiny compared to what lies within *us*.", 37

Feature-Agnostic Unaffected by salient features, 18

False Positive (FP) The error of tagging something as a positive match where it should have been negative, 128

Generation Grammar A set of grammatical rules for natural language generation, 11

Granularity The average size of tokens or entities used in a NLP task. This size affects the sensitivity of the system, 16

Hypernymy Semantic relation in which the semantic range of one word includes that of another (IsA), 32, 185

Hyponymy Semantic relation in which the semantic range of one word is included in that of another ($SUBSUMES$). Opposite of Hypernymy, 32

Isocolon A series of similarly structured elements having the same length. A kind of parallelism. Ex. "Veni, vidi, vinci.", 37

Lexical Choice In Computational Linguistics, the process by which a NLG system chooses synonyms, 12

Lexical Semantics Application of semantic analysis, 30

Monolithic System In Computer Science, a system where one large program handles most tasks, 13

Morphology In Computational Linguistics, the rules which govern word structure, 12

Multi-Agent System A computer system that distributes tasks among several specialized programs (agents), 11

n-Gram A sequential, ordered, and gapless set of tokens, 47, 163, 186

n-Gram Ovelap The number of consecutive tokens that are identical between two n-Grams, 47

Natural Language Generation (NLG) A sub-field of NLP which focuses on generation on human-readable text, 10, 185, 187

Natural Language Processing (NLP) See Computational Linguistics, 10, 45, 185, 186

Natural Language Understanding (NLU) A sub-field of NLP which focuses on understanding of human-readable text, 10

Optical Character Recognition (OCR) Electronic translation of scanned images of handwritten, typewritten or printed text into machine-encoded text, 163, 164, 174

Pronominalization Replacement of nouns with pronouns. See Coreference, 23

Rhetorical Anaphora Repetition of the same word or group of words at the beginning of successive clauses, sentences, or lines. Ex. "*This* car, *this* house, and *this* front lawn.", 37

Receiver Operating Characteristic (ROC) A graphical plot of the sensitivity (true positive false positive rate) for a binary classifier system as its discrimination threshold is varied, 128, 131, 134, 136–138, 141, 142, 156, 158, 159, 162, 167, 170

Schemes In rhetoric, figures of syntax, 27

Semiotic Analysis Application of Semiotics, 27

Semiotics The study of signs and communication, 187

Sentence Planner A framework that performs the Sentence Planning tasks of NLG, 11

Sentence Planning Also known as Microplanning, a sub-task of classical NLG which generally encompasses tasks that are involved with neither document planning nor surface realization, 187

Support Vector Machines (SVMs) A supervised learning method used for classification and regression, 21, 45

Synset Entities which comprise the basic units in the WordNet lexical ontology, 31

Syntax In Computational Linguistics, the rules which govern sentence structure, 12

Topic The subject matter of a conversation or discussion, 188

Topicality Arrangement by Topic, 23

True Positive (TP) The error of tagging something as a negative match where it should have been positive, 128

Text REtrieval Conference (TREC) A conference created to enable evaluation of information retrieval methods on large-scale datasets, 43, 45, 73, 163, 166, 167

Tuple An ordered set of non-repeating tokens that allows gaps between said tokens, 57

Extensible Markup Language (XML) A set of rules for encoding file in machine-readable form, 164