Data-Driven Creativity Enhancement through Word Association

by

Connor Hillen, **B.C.S.**

A thesis submitted to the
Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute for Computer Science
The School of Computer Science
Carleton University
Ottawa, Ontario
March, 2019

The undersigned hereby recommends to the
Faculty of Graduate and Postdoctoral Affairs
acceptance of the thesis

Data-Driven Creativity Enhancement through Word Association

submitted by **Connor Hillen**, **B.C.S.**

in partial fulfillment of the requirements for the degree of

Master of Computer Science

Professor David Mould, Thesis Supervisor

Professor Robert Biddle, School of Computer Science

Assistant Professor Hussein Al Osman,
School of Electrical Engineering and Computer Science

Assistant Professor Ahmed El-Roby, Chair,
School of Computer Science

Ottawa-Carleton Institute for Computer Science
The School of Computer Science
Carleton University
March, 2019

Abstract

Writing creative stories from a blank page is a challenging task, particularly in the modern game industry where dozens of writers can contribute to the stories and settings of a constantly evolving artificial world. We introduce a system which recommends interesting, evocative, and thematically coherent words to help creators write thematically connected stories. We combine principles from human creativity enhancement and computational creativity to build a creative assistant based on word association research. We show that careful corpus selection, filtering based on emotional sentiment, and promoting remote associations through paragraph scale segmentation can produce recommendations that promote creative goals better than alternative word association algorithms according to our creative word indicators.

Acknowledgments

I would like to begin by thanking my thesis supervisor, Dr. David Mould. His attention to detail, his patience, and his enthusiasm for this work was essential during this research. Regardless of the circumstances, his positive attitude and passion in our meetings would always leave me reinvigorated and ready to get back into the work. I could not have completed this work without his outstanding kindness, advice, patience, and support. His guidance has shaped my way of looking at computer science since my first year as an undergraduate student and has kept me driven to pursue this research.

I would like to thank the thesis committee for reviewing my work and providing me with valuable suggestions. Their input has improved this thesis and has helped me improve as a researcher. I would also like to acknowledge Carleton University, the School of Computer Science, and GIGL for their financial support and thank them for all of the amazing opportunities they have afforded me over these years. I would like to thank all of my friends, colleagues, and visiting scholars in the GIGL lab for their advice on this thesis, on research, and in academics.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Both artists and problem solvers use their creative skills to approach their work. When people create, they are solving difficult problems by coming up with new solutions based on their influential experiences. As the worlds created by storytellers become more disconnected from the real world, more complex, and more engaging, the consumer has also been given more control to drive their experience in these artificial worlds. In order to keep up with these added complexities, creators must seek out new and evocative influence to populate their worlds with content that separates it from other creator's worlds. We have focused on creators seeking to create exciting new worlds for their consumers where the details, thematic cohesion, and an abundance of content is required to be as engaging as possible. We review models of creativity and use these models to develop a system which takes in user provided terms and generate new terms which are thematically associated with the user input. We perform a topical segmentation of thousands of literary works and use these segments to associate terms in ways that emphasize literary connections. We then make use of features such as emotional sentiment to promote terms which will be evocative to creators and novel enough to help prompt new ideas.

We explore the creative process through the lens of video game developers and tabletop role-playing game writers. These domains require thematic cohesion, large amounts of content creation, and in the case of tabletop gaming, on demand improvisation as a reaction to changing circumstances. We explore the evolution of worldbuilding to help understand the processes that are in place when writing worlds for these domains and to better understand the motivations of the creators. Where early worldbuilding in stories established the reader as a passive observer of the happenings of fictional places, modern worldbuilding in the video game industry has

players taking full control of their new world, being able to experience the story in any order they wish and even to change the world itself. The challenge of building large, new worlds has moved worldbuilding from a solitary author to large teams of designers working across decades to expand upon their work. The complexity involved in producing massive game worlds makes it an excellent candidate for the growing field of *computational creativity*.

In his review of computational creativity research, Mitra recalls how early computer science research sought to build general problem solvers and in doing so to make systems that could creatively solve problems [37]. As the field has developed, it has trended toward building *creative tools* that help to bolster human creativity, rather than necessarily simulate it. While many games have made use of computer generated content for some time, research into computationally creative assistants to co-author content with a creator has only recently been explored. We use the principles from recent computational creativity works and principles developed in law and psychology for the purposes of creative problem solving to build a system that works with the creator to overcome the blank page problem. We have built a system that emulates parts of the creative process and traditional word association techniques, and show that non-creative word association algorithms perform poorly compared to a purpose-built system.

1.1 Understanding the Problem

When a creator creates they are tasked with making something that is both novel and successful in their domain [42]. Creators both intentionally and unintentionally take inspiration from their previous experiences when creating new things. Identifying associations between influences is a critical component of creativity [35]. For example, when a writer is coming up with a new story they might think about their past experiences and try to find ways to relate them to the current themes of the story. Associating these different themes allows the user to come up with new stories, and these new stories can prompt the creator to identify new potential influences and associations. Prompting new influences and associations has been the basis for most creativity enhancement research.

Previous creativity research has proposed different means of prompting new influences and associations; for example, there are wordplay methods which have the

creator change out words in the problem statement to identify overlooked influences, to put emphasis on different parts of the problem to find which influences are the most valuable, or to approach the problem with a different persona to view the problem from a different domain of influences [7]. We have focused our work on the former, with wordplay tasks and the building up of new influences being the ideal candidate for computational creativity enhancement [40]. These techniques help to satisfy the first aspect of creativity, novelty, but to help understand how we can help to create products which are successful to the target domain we have explored this problem through the lens of game writing, specifically open-world games in tabletop role-playing games and computer role-playing games.

When a player experiences a new world in a large, open-world game, they are often experiencing hundreds of disjoint stories that take place in a world written by dozens of writers. In order to maintain a sense of realism and history to the player, the world needs to follow a set of rules and themes that cohere the stories and relate them back other player's primary goal. Maintaining knowledge of these locations and trying to come up with a new location from a "blank page" makes it difficult to design the hundreds, or even thousands of locations that might be seen in a large-scale game today. New stories require lots of themes and imagery, making storytelling and worldbuilding very strong candidates for traditional word association exercises that have been used to enhance divergent thinking which can help to overcome the blank page problem. In addition to this, the word association algorithms that are successful in traditional natural language processing tasks build associations between words that are not ideal for creative purposes. We explored how stories were told across vast artificial worlds to better understand how stories are represented spatially in games.

Tabletop role-playing games typically involve telling brand new stories in artificial worlds entirely improvised with little preparation. Players are free to stray from the story that was initially conceived by the writer and can move into locations that have not been considered and have an expectation that the story will relate back to their previous adventures and be relevant to the area that they are exploring. In order to overcome this, various design tools are used to aid storytellers in tabletop role-playing games like *Dungeons & Dragons* [33,60]. These tools provide the creator with a visual representation of their new, *secondary*, world and can help to provide inspiration for new stories at a glance. For example, Worldspinner [33] will procedurally generate a

map and allow the player to either generate random events as a story starting point or to insert their own events on the map. When using their random events, a point will appear on the map with a random story, presumably taken from a list of pre-written stories. If the user wishes to add their own story, they can type in a text box to add notes for areas on the map. This tool makes use of both random generation and idea organization. These tools will typically focus on purely randomly generated content to assist with improvisation or organizing data in a way that allows the storyteller to recall information when it is relevant so that they can improvise relevant content.

To help creators create novel products, we make use of influence generation through word association techniques. To help create products which are successful in their domain, we have taken inspiration from some of the tools currently being used by tabletop role-playing game players. We found that many of the tools use random generation to quickly provide some inspiration to the storyteller while other tools will provide the storyteller with some way of organizing the stories and characters they have included in their game so far. They are usually organized spatially, with the goal of helping the creator recall relevant material when it is needed. We then result in creating a tool that helps generate influences when needed, but also to focus on relevant influences that help create new content while also making sure that relevant material is taken into account.

When players are exploring an unwritten area in a game, the storyteller might follow the following steps:

1. Consider some of the relevant themes: what did the players do last time they were there, what are they doing right now, and what are the themes of this area? Maybe the village is known for wine and merchants. Last time the adventurers were here, they were arrested by the merchant's guild. They have recently been following a shady aristocrat through different villages.

2. From there, the storyteller might try to connect the previous themes. A shady aristocrat and merchants might be dealing together. The aristocrat may be behind their arrest. The aristocrat is waiting for a wine shipment.

3. The storyteller can take those associations to build new themes. Crime, corruption, nobility, trafficking, collusion, conspiracy, wealth, shipping, gold

4. Finally, storyteller can associate those new themes to write potential stories for their players. Perhaps the merchants guild is corrupt and working with the

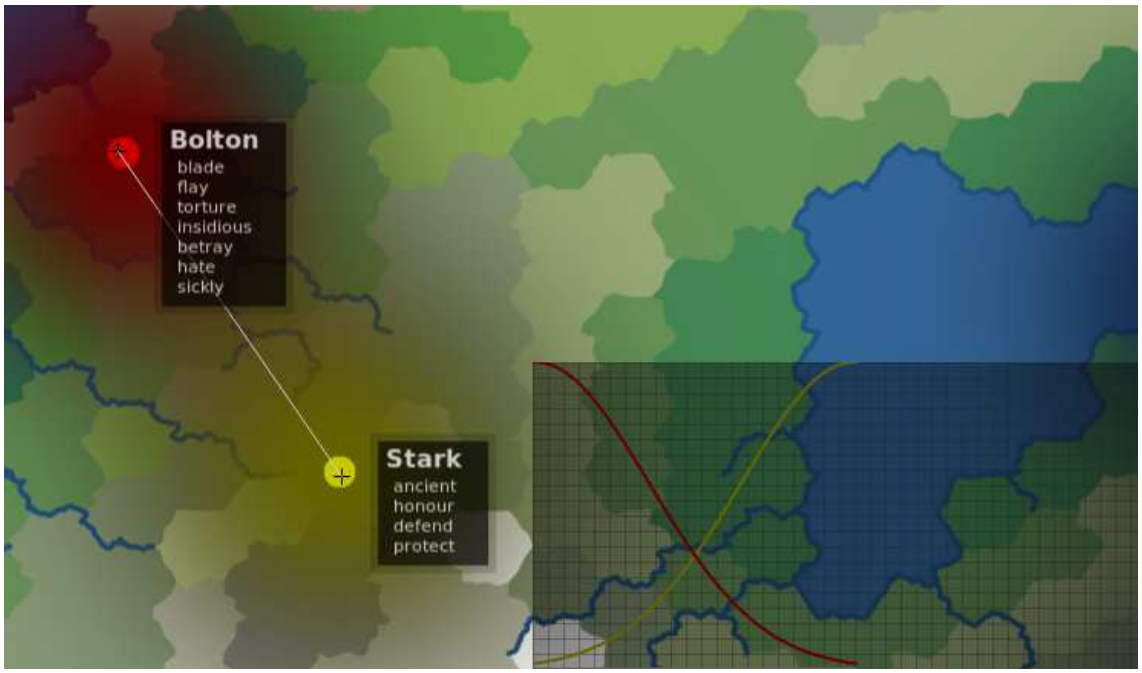


Figure 1.1: Early map prototype demonstrating two locations, represented by sets of themes. Graph in the bottom right displays the influence each location has on recommendations proportional to distance from the location centre.

aristocrat to traffic people in the wine shipments and the players can get their revenge on the merchant's guild by exposing them.

If the storyteller tracks their themes, we can develop a system to to generate new, evocative influences that they can use to write a new story. Early into the project we explored different ways that we might be able to aid in these middle steps. We explored simulation models similar to Sugarscape [16] to generate stories from simulation data and found that the events occurring in the simulation were not as important as the underlying themes that regions of the world embodies. Chris Crawford describes stories as "a mesh of interrelated ideas" [13] and describes that stories are about how all of the different information comes together to form a cohesive story. From this, we explored how multiple themes, represented as evocative words, can relate to one another to represent stories and settings within a world and how considered these stories might interact with one another spatially. Through the prototypes we built, shown in Figure 1.1, we were able to see how the themes of locations could be represented based on a few representative words for the purposes of our system.

In exploring these representations, it was clear that in order to aid in generating new stories on the map, we needed strong and interesting associations that were capable of prompting new, relevant ideas. Existing word association systems were not built with creativity in mind and did not spark the researcher's creativity in a way we had hoped. From here we designed a system built on creative principles to better understand how creative word associations differ from traditional natural language processing.

Our goal with this project is to help promote divergent thinking and overcome the blank page problem in large, open world games set in a secondary world. We do this by representing a location's setting as a set of thematic words and generating new wordlists that can represent events or settings in themselves by carefully associating the themes with evocative terms.

1.2 Contributions

In this thesis, we propose a system to generate word associations from a corpus of public domain literature that can aid in the production of creative stories. Our system contributes new ideas and systems that can computationally emulate different parts of the creative process:

- We explore how corpus selection determines the domain that creative influences come from
- We made novel filters based on a word's emotional impact and pre-existing word association data to remove influences that are not evocative
- We show that segmenting our corpus by paragraphs can produce strong, thematic associations that may not be obvious to the creator
- We make use of three different algorithms which all find influences in different ways

To evaluate our system against alternatives we developed some indicators that indicate when a word or set of words are not likely to support creativity. We used these indicators to evaluate over one hundred sets of recommendations and found that our system consistently outperformed alternative word association algorithms

that are not intended for creative purposes. The indicators were developed based on our creative goals and we believe they can act in place of user testing; however, user studies are required to validate these indicators as a heuristics for creative word associations.

We contribute to the field by exploring how we can bring together creative concepts with computation, specifically by using computational word association to enhance a creator's creativity as opposed to replacing it. We propose a set of indicators which we can use to evaluate lists of word associations for the purposes of creativity enhancement and designed a system that outperforms alternative word association methods according to our tests using these indicators. Our novel filters based on the emotions that words evoke to quantify how evocative a recommendation is allows us to aggressively filter out terms that do not spark creativity.

1.3 Overview

Chapter 2 discusses the various fields of work that motivate and influence our work. We start by discussing *worldbuilding*, the act of creating an artificial world for characters in a story to explore. The chapter continues to describe how games developed over the end of the 19th century to incorporate further story and worldbuilding, and how the modern complexity of storytelling games lends itself to the complex worlds developed in early fantasy literature. The evolution of the complexity of worldbuilding motivates the problem of coming up with creating entire worlds from a blank page in modern games. From there we discuss some research behind enhancing creativity in participants for the purposes of problem solving and artistic expression. Many exercises for creativity enhancement involve wordplay, and we continue on to discuss the field of computational creativity, creative assistants, and modern algorithms for associating words. Finally we discuss some similar work that use computational word association techniques for creative purposes.

Chapter 3 starts by providing an overview of our word recommendation system. It presents a five-part design review of the system and explains some of the experimentation that led to the final design as well as details regarding the algorithms and data that was used. It continues to discuss some simple implementation information that may not be immediately apparent when first approaching the design, including some new Python libraries that proved valuable when processing our data. Finally

we conclude the chapter by discussing the design of our recommended word quality indicators and the tool we developed to evaluate recommendations using these indicators.

The next chapter discusses some results that we obtained when experimenting with the different components of our system. We compare one of our configurations to alternative methods using our quality indicators and discuss some of the issues with our system and others and speculate at the next experiments required to help validate our indicators and measure the impact of each system component on the final recommendations.

In the final chapter of this thesis, we review the goals of the project and our contributions to the field. We then discuss various opportunities for continued work and some of the results from exploratory work that can be explored more thoroughly.

Chapter 2

Background

2.1 Introduction

This thesis explores the problems in modern storytelling when entirely new worlds are being created for people to explore and interact with in videogames. We review the works that motivate this problem and propose a solution combining principles from multiple fields, including the study of creativity in psychology and literature, the study of creativity enhancement through law and and psychology, the techniques for working with computational assistants and generative art that is researched in computational creativity, and the basics of natural language processing from computer science. By combining some of the principles in each of these fields we are able to build a creative assistant intended to help writers and game developers be more creative when designing new worlds.

How creativity is defined and practiced has been in debate for the past few centuries. While the concept of creativity had originally reserved for the divine and later accepted for art forms such as poetry, it was not until the beginning of the 20th century that creation became the accepted term for many different fields [54]. Wider adoption into the arts and sciences demanded new research into creativity enhancement. Many different disciplines seek to enhance creativity to help solve problems, start businesses, evaluate difficult situations or evidence, and produce new and unique art. Game development lies at the crossroads of problem solving and artistic expression. Players of games are typically taught to solve problems and make meaningful choices all the while realizing the artistic intentions of the game developer. The world created by the game developer not only determines the possible narratives and experience that the player is intended to have, but it also builds a framework of rules that

the player must work within to solve the various problems that make up the primary fun of the game.

Games can be distinguished from other media by the interactions and choices that players make. Players find fun in games by experiencing the unknown, by learning, and by overcoming challenges [25,30]. This drive for consuming unknown worlds combined with learning novel skills to overcome challenges typically requires the developer to create a new world which is familiar enough to understand quickly, yet functions in a novel way that lets the player learn and explore. *Worldbuilding*, the process of constructing a new world or universe, poses a particularly difficult challenge when trying to maintain coherence between places, events, and concepts. Video games have taken inspiration from many sources to evolve the art form. While worldbuilding has been a very popular component in videogame narrative since the field's infancy, worldbuilding in other media has only truly evolved in other media since the widespread understanding of art as a form of creation that was adopted in the 19th century. The importance of narrative and systemic novelty in games poses many unique challenges which include the emphasis on worldbuilding, but industry pressures also contribute to the difficulty of creating video games.

Games often require tight deadlines, hundreds of employees, and can include dozens of writers on staff. Games with constructed worlds can sometimes have thousands of characters, locations, and stories to be experienced by the player with limited control over the order of events that the player will experience. In this section, I describe how worldbuilding developed in media and how it found its place in game development, the importance of creativity and storytelling in games, some established methods for enhancing creativity in various disciplines, and the use of computation in creativity enhancement. Finally, this chapter describes various methods for word association, how they are used outside of creativity enhancement, and explores similar work using data and word associations to help create better stories.

2.2 Games, Worldbuilding, and Worldness

For thousands of years, writers have created imaginary new places to entertain audiences. Readers have always loved hearing stories from far away lands, including the mystical islands found in Homer's *Odyssey* and the impossible space travels found in the 1600's with works such as Kepler's *Somnium*. As the imagined places of early

literature became more detailed and sophisticated, authors like Tolkien and Baum began creating entirely new worlds such as *Middle-Earth* and *Oz*, respectively. These stories, though mystical and impossible, were often presented as an entire reality within a *secondary world*. A secondary world is a new world which has been written by a creator with varying degrees of separation from our own world. Secondary worlds lie on a spectrum: truly secondary world might have almost no ties to the real, *primary* world at all, such as Middle-Earth, whereas some secondary worlds may be more deeply connected to the primary world, such as the Dorothy's travels between Kansas and Oz. Where Oz grounds the reader in the real world before introducing the secondary world of Oz, Tolkien puts the reader into a new universe, seemingly distinct from our own.

Worldbuilding has evolved over thousands of years, from old traveller's tales with passive observers to modern fantasy worlds where the characters are born in and interact with their entirely made up worlds. Games have continued this evolution, not only letting the consumer read the story of a character which interacts with the secondary world, but allowing the player themselves to interact with, influence, and even create their own secondary worlds. As the complexity of these authored worlds has increased and the complexity of the consumers interactions with the worlds have increased, the task of building and maintaining a secondary world over the course of decades in video games has raised new problems in collaborative creativity. This section reviews the motivations of worldbuilding, the nature of stories in games, and provides some additional context for the motivating problems of this thesis.

2.2.1 Worldbuilding in Traditional Media

The concept of *worldbuilding*, of creating or "subcreating" new worlds has been explored by literary theorists over the past few decades with the rise of franchise and transmodal entertainment, as discussed by Wolf in his book *Building Imaginary Worlds* [59]. Worldbuilding involves creating a *secondary world*, a world somewhat detached from our own *primary world*. Secondary worlds are defined by how disconnected they are from the primary world that we live in. During the Age of Sail, stories were told of fantastic islands that the average reader would never be able to venture to, but that actually existed in our own world or supposedly existed within our world. When more of the world began to be mapped, secondary worlds took the form of divine places, like Dante Alighieri's circles of hell in *Divine Comedy*.

A few features of secondary worlds tied early works together: The characters were often observers to these new worlds, rarely interacting with or changing the secondary world in any meaningful way, and the tails of the secondary world must have been able to make it to the primary world in some way. As Wolf notes, the extensive mapping of Earth's islands led early science fiction to write traveller's tales which took place on other planets, typically ones which were very nearby to Earth. It was not until Defontenay's *Star ou Ψ de Cassiope* in 1854 that authors would write about far-away planets that we could not explore, only because the story is claimed to have been found in a meteor and translated from an alien language. Around the same time as Star, the new genre of fantasy was evolving. The rules of these fantasy worlds were not very strict, and the authors tended to lean into the dreamlike nonsensical states that could exist. Many nonsense worlds, such as Wonderland in Lewis Carroll's *Alice's Adventures in Wonderland* began to emerge, where the new world was entirely imaginative. This inspired writers like Frank Baum to explore nonsensical worlds bound by rules and grounded with the primary world with his series surrounding the world of *Oz*. Wolf considers the Oz series to be the first major series to emphasize the secondary world over the individual characters. Writers like C.S. Lewis and Tolkien refer to Oz as one of their primary influences in their childhood worldbuilding and eventually their worlds of Narnia and Middle-Earth.

The worldbuilding of Tolkien and Lewis in the 20th century launched a new trend of storytelling. The main characters of a secondary world story no longer needed to be explorers from the primary world, but were instead citizens of the secondary world. Characters were more free to explore and interact with the world than ever before and all forms of media began to use worldbuilding to build lucrative franchises and cult classics. Rather than only relying on interesting characters, franchises would benefit from having interesting and unfamiliar worlds that many characters and stories can occur in. The rise of secondary worlds where the characters are first class citizens of the secondary world seems to coincide with the rise of complex worldbuilding in games in the 20th century.

2.2.2 The Rise of Worldbuilding and Storytelling in Games

Modern story in games has its roots in *tabletop wargaming*. Early wargaming included games such as Chess, where players control units and employ strategies to control territory and defeat the opponent using abstractions of military strategy. *Kriegsspiel*,

German for "war game", was an early variant of Chess intended to model the essential concepts of war [26]. Later in the 19th century, a Kriegsspiel was developed using military miniatures and was intended to train and educate rather than entertain. By the end of the 19th century, wargames were starting to move out of military training and entertainment and began to be played by civilians. After World War II the interest in commercial wargames piqued and many wargame variants began to be produced and distributed. The popularity of immersive war games and the rise of worldbuilding in fantasy led the creation of the first *tabletop role playing game (RPG)*, *Dungeons & Dragons (D&D)* by Gary Gygax in 1974 after the success of his medieval fantasy wargame *Chainmail* in 1971 [46].

D&D moved away from traditional wargaming by having players control only a single character rather than an entire battalion or military. The players were expected to play through a series of short stories which comprise a larger campaign. Similar to the motivations of early 20th century franchises, tabletop RPGs required the player to have many interesting stories within a single world. Over the years, hundreds of official and unofficial supplementary books have been written to describe the various worlds, creatures, rules, and pantheons of the D&D universe to build a franchise spanning hundreds of novels, video games, and films.

2.2.3 Stories in Games

While games have been around for thousands of years, they have traditionally been examined through the lenses of rules and systems. Games are often non-linear, made up of the explicit and implicit choices of the player. The introduction of player choice leads to novel narrative structures that have not been explored by previous mediums. Some of the different categories of narrative in games include [44]:

- **Pre-established Structure**: Pre-established structures present the player with story as they approach and achieve various checkpoints. The story in this structure is presenting linearly, similar to traditional mediums, giving minimal control of the narrative to the player and a lot of control to the narrative designer.

- **Discovery Narrative**: In this model, players are presented with some overarching pre-established narrative but are also allowed to momentarily explore

smaller side-narratives. This model allows the player to discover their own narrative within the established story. An example of this can be seen in the game *The Elder Scrolls V: Skyrim*, where players are presented with a mostly linear storyline that they can choose to complete, but are allowed to progress through that story whenever they choose. At almost any point in the game, the player can freely explore the world of Skyrim and encounter smaller stories, or *sidequests*, which may slightly alter the overarching story. These smaller sidequests are typically used to support and contextualize the overarching story in some way.

- **Sandbox Narrative**: The sandbox narrative emphasizes the emergent interactions of systems within the game to enable unique and personalized stories for each player. These games may still have a discovery narrative in them, but players are more likely to have personalized and unexpected narratives arise through play. Elements of sandbox narratives can be seen in discovery games like *The Elder Scrolls V: Skyrim*, typically revolving around stories of lesser importance.

Playing Dungeons & Dragons inspired early programmers to develop some of the first computer role playing games with discovery narratives, *Colossal Cave Adventure* and *Ultima*, in 1977 and 1981 respectively. While the definition of open world game varies, both Ultima and Colossal Cave Adventure are often credited as being the first open world computer games that resemble the modern definition of open world exploration [22, 39]. In addition to being open-world, Ultima also had an entirely constructed universe that was built upon across multiple games, similar to D&D. The importance of replayable, explorable, and open worlds became prevalent in the gaming industry.

In stark contrast to the early development of games like Ultima which typically only involved one programmer working over the course of a few months, modern games such as *The Elder Scrolls V: Skyrim* have nearly 100 developers working for years at a time; of those developers, over 20 are credited with roles which involve writing and designing levels [1]. Skyrim, only one game in an interconnected series, is estimated to have over 300 quests, 500 locations, 800 written stories, and 1000 interactable characters [10].

When players are interacting with these complex worlds, it is important that they are able to understand the large amounts of story at a glance [31]. Critics have

described games as being "spatial narratives", where there is increased importance placed on the placement of characters, symbols, and stories around the geography [27]. In a sandbox or discovery narrative, the designer must find ways to clearly make places and cultures distinct, identifiable, and coherent.

With the ever growing scale and complexity of games and their stories, it is important to have teams which are empowered to come up with creative ideas for systems, stories, and to solve problems that may occur in the game's creation. Understanding creativity and how we can be more creative is important to creating bigger and better games.

2.3 Creativity Enhancement

There have been many techniques developed to enhance creativity that have been built on developing understandings of the different processes involved in novel creation. This section reviews some definitions of creativity, discusses studies which review and define the creative process, and discusses how people have used word association and other exercises to be more creative. We will also examine some existing tools that help game creators come up with better stories.

2.3.1 Defining and Enhancing Creativity

There is a need for creativity when solving problems, managing a company, writing, practicing law, teaching, and various creative arts [40]. There have been many attempts to define creativity, including E.P. Torrance's definition of creativity as "the process of becoming sensitive to problems, deficiencies, gaps in knowledge, missing elements, disharmonies, and so on; identifying the difficulty; searching for solutions, making guesses, or formulating hypotheses about deficiencies, testing, and retesting these hypotheses and possibly modifying and retesting them; and finally communicating the results" [56]. In addition to this, Raymond Pfeiffer proposes that a product of creativity is "a piece of work which is first to a significant extent new, original, and unique and second shows a high degree of success in its field" [42]. These definitions were included in a review of creativity research by Götz [19]. While the exact definitions are up for debate it acts as a fine starting place to see creativity enhancement as a way to help account for gaps in knowledge, help in creating new and original solutions, and to help in producing successful results.

Methods for enhancing creativity have been studied for some time. After a study of 100 participants, Muller was able to find a common process for developing creative products [40]. According to Muller, every creative product starts with some motivation and a *thorny problem* — a problem which can not be solved purely through reasoning alone. Understanding both the motivation and the thorny problem results in a *driving force*. Once the driving force has been realized, Muller states that the mind seems to be primed to seek out any possible links to the problem and argues that this priming is the most crucial factor in enhancing creativity. To help find links, the problem will also be given constraints and influences. These influences are typically the domain of computer aided creativity. It is through the process of priming with the driving force, understanding the constraints, and linking concepts through various influences that leads to the creation of many products, eventually leading to the final creative product.

Many methods for enhancing creativity include rearranging the problem, role playing, or introducing new concepts and being forced to make connections [7]. These methods take advantage of the priming that occurs once the creator understands the driving force and is trying to find connections. In a review of creativity enhancement techniques, Brown categorized various methods to build connections and help better understand the driving force; important to this thesis are Brown's categories of *wordplay* and *word clustering*.

Wordplay specifically looks at how individual words can help to build new links. Creators are tasked with writing out their problems and forces. In wordplay, creators may look at a random word and find a way to incorporate that word into their driving force. This allows creators to find new influences that they may have missed. When removing or modifying words, creators shift perspective on the problem and may find that some influences are not as important as they seem.

Word clustering is similar to word addition tasks. Creators are asked to write out all of the words that come to mind after recognizing the driving force. The words should not be written in any particular order. Creators then try to find connections between the words, helping them to both understand what their influences may be and to visualize the different ways those influences are connected.

Once the creator has a clear understanding of the driving force, wordplay seems to be able to enrich their ability to think creatively and to create new things. The connection between word associations and creativity has been actively explored since

the 1960's [35] and has interesting implications for how computers can aid in the creative process.

2.3.2 Word Associations and Creativity

We discussed how word association tasks have been used for creativity enhancement, but our research requires us to explore how certain associations may be more influential than others for creative tasks. There are many classes of word associations [52] that associate terms, which we can take advantage of for creativity enhancement; some of these relationships include:

- Class or hypernymous inclusion, where a word is a class of another, such as *vehicle* which is a hypernym of *car*
- Coordinate relationships, where two terms share hypernyms, such as *man* and *woman*, which are both of multiple similar classes
- Meronymous inclusion, where an entity contains other entity parts. These parts can be components, members of a collection, or a portion of the parent entity, for example. These could include *face* and its part *nose*
- Case relationships, where the word has various attributes or actions associated with it. For example, *bell* could be related by case with the verb *ring*.
- Antonymous relationships, where a word has the opposite meaning of another, such as *happy* and *sad*
- Synonymous relationships, where a word has the same or similar meaning as another, such as *happy* and *joyous*
- Paradigmatic relationships, where a word can be swapped out with another word of the same classification, such as *dog* and *cat*, or *the man* and *the woman*.
- Syntagmatic relationships, where one word might follow another, such as *drive* and *car* or *kick* and *foot*.

By better understanding these relationships, we can gain a better understanding of what causes one word to be a better influence for creativity than another. Aside from enhancing creativity, a person's ability to associate words has been a psychometric

test to evaluate human creativity since Mednick's work in 1962 [35]. The remote associates test, or RAT, presents the participant with three seemingly unrelated cue words and asks that participant to come up with a word that connects all three together. The task is intended to be challenging, but computers find the RAT to be trivial [55]. While Mednick believes that the test can measure a participant's creativity, Runco and Acar argue that associative tests fail to measure originality, a necessity for creativity [2]. When evaluating creativity, Runco and Acar discuss how divergent thinkers will be able to find associations to the initial task which are remote, but not entirely irrelevant.

In their research, Runco and Acar review works that show words used together will typically elicit one another, building a lexical neighbourhood, and that this neighbourhood can be used to measure close, remote, and irrelevant associations. They propose that it is important to understand these distances to better evaluate creative thinking and that association networks can help in producing objective judgments of creativity enhancement tasks.

Understanding the different types of word associations and how remote associations need to be for creativity is crucial for building a system which aids in literary creation. Research shows that word association can aid and measure creativity; however, it is important to keep associations in within a careful neighbourhood to ensure that associations are remote enough to elicit novel ideas rather than associate directly. We can use these principles to develop our tools, but first we examine the different tools that are already in use to support creativity in games.

2.3.3 Creative Tools for Worldbuilding in Games

There have been many software solutions to aid in the complex task of worldbuilding in both tabletop role-playing games and computer games. Some methods focus on random content generation while others are simply tools to help the creator keep track of the game world. Various techniques for generation have been used, including pure randomization and generation from analyzing datasets [14]. These generative methods are typically used to reduce the amount of content required to make the world feel more content rich and to help the creator come up with new ideas for interesting side stories.

While these methods are valuable for helping the creator keep track of information and generate random content, they do not try to aid in the creation of new creative

artifacts that cohere with their existing game world. What these tools outline to us is that creators can find value in tools which help to inform story by keeping spatially relevant information available at a glance and that creators are interested in using computers to generate content that they can use to help with blank page problems.

What these tools seem to be lacking is an ability to contribute to the creative process in a meaningful way. These systems either do not contribute to the creator, having them submit all or most of the data, or contribute in a purely random way which can lead to many irrelevant associations. In the next section we describe how computers can be used to produce creative results and contribute to the creative process in a meaningful way.

2.4 Computational Creativity

There are many different ways that computers can help to produce creative products and various perspectives on the place of a computer in the creative process. Many computer games make use of procedurally generated content in increase the replayability of their games and, in some cases, reduce the overall production requirement. These can come in the form of level generation algorithms [53, 61], creature generation, galaxy generation [24], culture generation [28], history generation [3, 18], and more recently dialogue generation [48].

2.4.1 Generative Art

Computational creativity has explored the generation of art in many disciplines. Many of these generative systems use a set of rules that are either compiled by a human or built automatically by analyzing data. ANGELINA is one such system [12]. ANGELINA designs simple platformer games by evolving separate components of the game simultaneously and simulating gameplay. In later iterations of ANGELINA, the system was able to produce games in 3D. Similar to our work, the new version of ANGELINA reads a seed word from the user and uses word associations from WordAssociations.net to enhance the theme [11]. While procedural generation is prominent in games, computationally creative generation can be seen in many artistic disciplines.

The Painting Fool is a generative art program that was intentionally designed to not be a tool for artists, but an artist in itself [9]. The system uses machine learning and various methods of learning to create evocative art that has been showcased in

art galleries. Other researchers are using narrative theory as the basis for simple story generation [12] and using data analysis to generate poetry [55].

While these generative art pieces are interesting and help us better understand the creative process, they are not able to replace artists at the moment. Most generative art is highly dependent on the input and work like generative poetry still need more work to be indistinguishable from human writing. We use these generative examples as a basis for understanding how we can see the creative process in terms of algorithms and better our understanding of the requirements of generative art. Both Toivonen's generative poetry [55] and ANGELINA's generative games [11] are starting to make use of word associations in generative art in different domains, and while the goal of this thesis is to bolster an author's creativity we also understand the need for better creative word associations in generative art.

2.4.2 Computationally Creative Assistance

In addition to purely generative techniques, computationally creative assistants are being used to collaborate alongside human creators rather than simply generating the final result. Collaborative AI typically tries to replace a human in traditional group creativity enhancement and enhance it by bringing in different domains [29]. When describing creative assistants, Koch notes some methods for collaborative creativity which are very similar to Muller's individual model [5] and uses these methods of collaboration to describe the potential use of AI during the creative process.

Initially, the creator must find their driving force and frame the problem. The system can assist in this by asking questions that hone in on the motivations. The system can use the participants responses to suggest some ideas, refer to similar projects, or infer information about the problem and present it. This lets the participant adapt their understanding and provide feedback to adapt the systems understanding. Next the participant will explore the problem space, and the system will suggest information both in and out of the relevant domain. Through these suggestions the participant will help the system to establish constraints and visa versa. Finally the system and the participant can begin to come up with solutions and the system can highlight specific components.

Koch's model has similarities to Brown's categorizations of creativity enhancement techniques and Muller's model of creative processes. The system aids in discovering

new influences through the addition of new concepts, builds links by explicitly searching for related information, and performs emphasis shifting by highlighting different components which helps in refining issues with the final creative product and give the participant a better understanding of the driving force.

Combining these models of creativity enhancement and collaborative AI gives us a starting point to understand the design requirements of our system. These tools allow us to review current computer models for associating words through the lens of the creative process and provides us with the insight needed to understand the requirements of a creative assistant in our chosen domain.

2.5 Computational Word Association

Computational word association is a well explored topic in natural language processing. Relating words to one another allows researchers to statistically evaluate the importance of a word as it shows up in one document and in doing so identify what separates one document from another. This has many applications, including automatic summarization [15], search engines [38], and sentiment analysis [51, 58]. These models are typically finding relationships which are based on meaning or on usage: for example, the word *flower* is typically nearest to types of flowers and a country would be nearest to other countries. While these relationships are excellent for their intended applications, they are also obvious to people and do not reveal new and interesting influences as required in the creative process [43].

We explored some of the traditional models but our primary focus was not in achieving highly accurate syntactic relationships or finding synonymous terms. We instead focused our exploration on how we could use simple association models and rely on the rest of our system to ensure words are associated by domain, by theme, and that the words we recommend are evocative. This section explores some of the methods that have been used to associate words, how sentimental analysis provides us with a starting point for detecting how evocative a term is, and discusses a related word association system.

2.5.1 TF-IDF

TF-IDF, or *term frequency-inverse document frequency*, is a common weighting scheme that provides a score to words based on how important they are to a document within a corpus. A word is considered "important" to a document if it occurs more frequently in the document than in other documents in the corpus. Before any kind of weighting scheme can be used to rank associations between words words, the documents should be converted into a representation that is easier to process, such as the bag-of-words (BoW) representation.

A BoW representation is simply a counter representation of the words in a document that is part of a greater corpus. To convert a document into BoW the entire corpus is reduced into an indexed dictionary containing all of the words in the corpus. This dictionary can then be used to represent documents as vectors. Each document can be represented as an N length vector, where N is the number of unique words in the dictionary. Each index of the vector is mapped with one unique word in the dictionary. Each document is then represented as an N length vector where each value in the vector is mapped with the number of occurrences of each word in the document. Figure 2.1 shows a simple BoW representation for two different documents in a corpus.

Before representing the document as a BoW, there is typically some pre-processing which either removes some words, converts words into a different form, or groups words together to preserve additional context when represented as a vector. Stop word removal is a common first pre-processing step for most text processing tasks. Stop words are words that contribute little meaning to a sentence on their own: for example, *the* or *is*. These words are typically removed because they are very common yet provide little information. Words can also be reduced to all lowercase and have punctuation removed to remove duplicate entries that are simply used or formatted differently. To further remove duplicate words, words may be changed to a common *lemma* form, for example *rang, ringing,* and *rings* may become *ring* or *mice* may become *mouse*. All of these remove some information about the word, such as the tense or any important capitalization, and can not be applied universally for every problem. The BoW representation itself removes a lot of contextual information by getting rid of word ordering. To retain some context, some problems may make use of *n-grams*, typically bigrams or trigrams, where a token contains n consecutive words rather than a single word. An example bigram bag-of-words can be seen in Figure

Doc 1: Tommy wants to run with the dog, but the dog does not want to.
Doc 2: Tommy wants to ride a bike.

Dict	Doc 1	Doc 2
Tommy	1	1
Wants	2	1
To	2	1
Run	1	0
With	1	0
The	2	0
Dog	2	0
But	1	0
Does	1	0
Not	1	0
Ride	0	1
A	0	1
Bike	0	1

Figure 2.1: Example of a bag-of-words representation for two documents in the same corpus.

2.2.

The BoW model serves as a starting point to compare documents as vectors and perform statistical operations on them. TF-IDF can be easily calculated on a corpus that has been reduced into bag-of-words, either with single word tokens or with n-grams. The basic TF-IDF formula can be represented as:

$$\text{TF(term, document)} = \frac{\text{\# instances of term in document}}{\text{\# terms in the document}} \tag{2.1}$$

$$\text{IDF(term, corpus)} = \frac{\text{\# documents in corpus}}{\text{\# documents containing term}} \tag{2.2}$$

$$\text{TFIDF(TF(term, document), IDF(term, corpus))} = \log_e(IDF) * TF \tag{2.3}$$

The term frequency is often normalized to account for different document lengths and the taking the logarithm of inverse document frequency prevents the value from

Doc 1: Tommy wants to run with the dog, but the dog does not want to.
Doc 2: Tommy wants to ride a bike.

Dict	Doc 1	Doc 2
Tommy want	1	1
want to	2	1
to run	1	0
run with	1	0
with the	1	0
the dog	2	0
...	...	...
to ride	0	1
ride a	0	1
a bike	0	1

Figure 2.2: Example of a bigram bag-of-words representation for two documents in the same corpus.

completely overwhelming the term frequency. There are many variants on TF-IDF that are used in different situations [34], such as using binary term counts when it only matters that a term appears in the document and not how many times, applying a weight to the TF or IDF values, or normalizing the values based on the maximum corpus frequencies. All of these are used to calculate the specificity of a term [50] and the importance of a term to a particular document. If a word appears frequently in a document and infrequently in a corpus then it is considered important to the document and the resulting TF-IDF of the term in the document will be high.

While TF-IDF is typically used to represent a term's importance to a *document*, if we treat each word in a corpus as a bag-of-words document where the values are derived from collocated terms we can perform a TF-IDF that ranks the importance of each word in the corpus to every other word. TF-IDF is valuable for our work as it measures *importance* as opposed to measuring *syntax*.

2.5.2 Distribution Models

Other techniques look at documents as distributions of words and attempt to compare the two distributions. These approaches assume that a document is "generated" according to some statistical distribution and that two documents can be compared

by seeing how likely it is that their BoW were generated by the same function. Comparing the likelihood that two documents were generated by the same function can be done simply using methods like chi-squared [8], but a lot of success has been seen using the log-likelihood ratio (LLR) [15,38,55]. LLR provides a measure of how likely it is for a word, w_0 to appear in a BoW vector alongside another, w_1, compared to the probability that the words will appear independently. Written formally, we have two hypotheses:

$$\begin{aligned} H_0 &: P(w_1|w_0) = P(w_1|\neg w_0) \\ H_1 &: P(w_1|w_0) \neq P(w_1|\neg w_0) \end{aligned} \tag{2.4}$$

We can then take the ratio of the likelihoods of these hypotheses, $LR = (\frac{L(H_0)}{L(H_1)}$ and use that to find the LLR, typically with $-2 \times log(LR)$. The LLR provides us with one statistical measure of co-occurrence in the corpus. Another similarity score that is often used for text processing using BoW vectors is cosine similarity.

The cosine similarity represents the cosine of the angle between two vectors and the cosine distance is simply $1-$cosine similarity. This is often used at the document level, comparing BoW vectors to one-another using a measure such as TF-IDF weightings rather than just collocation counts [4]; however, this can be applied to other vectors, such as vectors of word collocations across a corpus. The cosine similarity between two vectors A and B can be found using:

$$\text{similarity} = \frac{A \cdot B}{||A||||B||} \tag{2.5}$$

The cosine similarity provides a measure of similarity or dissimilarity of the usage of two words, providing different results to LLR which measures the likelihood of two words co-occurring. Machine learning techniques typically make use of these types of similarity scores to create vector representations of words [6, 36]. These methods make use of large datasets, such as *Google News* or *Wikipedia* in order to obtain more statistical information for these calculations.

2.5.3 Sentiment Analysis

Sentiment analysis uses word associations to approximately quantify the overall attractiveness or averseness (valence) of a document. In addition to this, more specific models are being built to identify the specific emotions that words are most commonly

associated with [51, 58]. These models are typically built by asking participants to recall a story which elicits a particular emotion [49] or by asking participants to read a document and record what emotions they felt after reading it [51].

On their own, these word association models succeed at finding close relations between words and are able to discover various interesting and useful features. What is interesting is how people have made use of all of these various technologies to stimulate creative growth as opposed to tasks such as summarization.

2.6 Similar Work

In this chapter, we have reviewed storytelling in games, how we might be able to enhance creativity with and without computational assistance, how wordplay can play a critical role in creativity, and how many computational word association networks function today. From the beginning, our research sought to bring many different ideas together to explore how all of these components can come together to help creators tell better stories. In this section, we will briefly explore some closely related works that try to combine some of the topics we have discussed previously.

2.6.1 Word Association Network

Used as a network to test divergent thinking in Runco and Acar's work [2], *Word Association Network* is a network developed by Yuriy A. Rotmistrov. The network claims to make use of literary work, similar to how we make use of a literary corpus. There is limited information regarding the algorithms used to discover associations outside of its use of literary works [47] to build associations. Similar to this thesis, Word Association Network also seeks to help writers, as well as various other professions and makes use of a carefully selected corpus of literary works. While the system can produce some interesting results, the blackbox nature of the system and the lack of flexibility leads us to continue exploring the underlying mechanics of creative word associations.

2.6.2 Creative Word Associations for Generative Art

Making similar assertions to our work, Toivonen et al. explored the use of word association algorithms in the production of poetry and creative evaluations [55]. When

comparing their results to the RAT, Toivonen et al. were able to find that simple algorithms could easily outperform human participants [20]. In seeing these results, they then sought to perform a more generalized version of the RAT to find connections that were less obvious. While they initially used the popular Google bigram dataset to pass the RAT, Toivonen et al. used sentences from Wikipedia to find more remote associations between words. They successfully performed well in their designed test; however, the Wikipedia associations cover a very broad range of topics. While we independently came to some similar principles as Toivonen et al. we also find our work emphasizing the use of these word associations for creativity enhancement rather than creative generation and focus our efforts on generating evocative associations.

Similar to Toivonen et al. this thesis uses a larger window size to build more remote collocations than bigram analysis will typically provide. We use a corpus of literary works similar to Word Association Network to provide links that are commonly found in creative works, and make use of emotional analysis of words to promote evocative associations. We are inspired by the spatial representations of game stories that many of the current tools for game creators make use of and use the work of computational creativity to generate interesting and evocative lists that have their place in the theories of collaborative creativity enhancement. While this collection of techniques can be used in many fields, our work focuses on storytelling in games due to their complex, spatial, and collaborative nature.

Chapter 3

Methodology

3.1 Overview

Our word recommendation system takes thematic elements from the user in the form of a wordlist and uses a pipeline, inspired by the creative process and creativity enhancement literature, to generate new influential themes that can support storytelling within the creator's existing work. By parsing a literary corpus for word associations, our system acts as an external actor that stimulates the discovery of new influences derived from historical literature to better define the creator's problem from a literary perspective. The system uses stimuli terms that represent locally relevant themes in order to ground the influences in the creator's world. The system ranks the influences by novelty and by relevance in order to maintain recommendations that contribute to the story in a clear and meaningful, yet unexpected way. The resulting wordlist is then processed further to remove any poor quality recommendations before presenting them to the user.

This chapter begins by describing the system design, including some discussion on alternative designs that were briefly explored. While this section is described with respect to recommending a set of words from a single stimulus, the next section describes how we designed recommendations to generalize across multiple stimuli and the problems introduced by using multiple stimuli. The chapter continues to briefly describe some implementation details, including information on the libraries and data structures that were used. The chapter concludes by describing the design and justifications of our system evaluation.

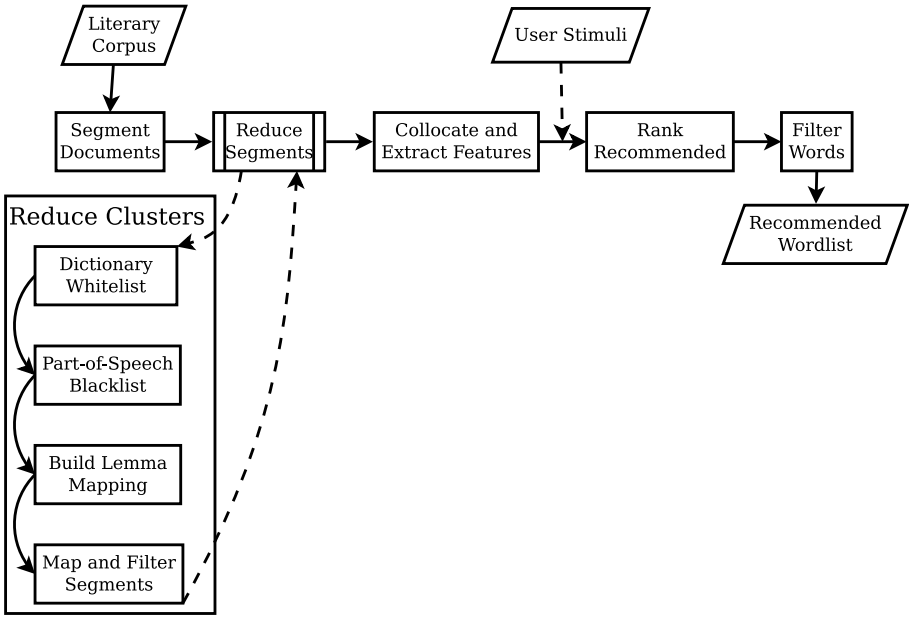


Figure 3.1: Overview of the main system components.

3.2 System Design

The word recommendation system can be broken into five primary components, illustrated in Figure 3.1: dataset segmentation, segment reduction, collocation and feature extraction, ranking, and runtime filtering. The system begins by converting a corpus of freely available novels into smaller, unordered segments that represent the basis for how words can be associated. The corpus selection defines the creative domain of the system; words associated from a literary corpus will provide more literary influences whereas associations built from news articles will produce more journalistic associations.

After segmentation, the segments are reduced and represented as a bag-of-words vector. At this phase, we reduce the segment by filtering out many poor quality words and convert the words into a neutral base lemma form. The lemmas help us to connect concepts better with a smaller corpus and help connect user provided stimuli to our vectors.

Once the segments are reduced and represented as a bag-of-words vector we extract some features, such as term frequency and inverse document frequency, for later parts of the system. We can begin creating collocation vectors for each word, resulting in an NxN collocation matrix which can be further processed into a weighted collocation

matrix if using our weighted scheme.

Once the collocations are built we can generate recommendations for user provided stimuli. We do this by either taking distances between the collocation vectors or by returning the highest weighted terms of our weighted collocated matrix for the provided stimuli vectors. These output large lists of recommendations which go to the final phase of our system.

Before returning the list of recommendations that were ranked in the previous phase, we pass the recommendations through one final filter. This final filter removes words based on our measure for emotional intensity and how frequently they occurred in the corpus. These filters are applied at runtime which means the user can adjust how aggressively the system filters out terms.

With each component of the pipeline added, the difficulty of performing a rigorous validation becomes much more difficult and further analysis of the design would require more time and parameter tuning of each component. We present some of our final design decisions, but note that we have not explored all combinations and continued work would focus on comparison to the other methods we describe.

Before the system can be used the dataset must be built from a suitable corpus and then reduced into segments that can inform our collocation data. Our work explored the use of different textual corpora and how the different corpora influenced creative recommendations and how different segmentation techniques influence the relevancy of word recommendations.

3.2.1 Building the Dataset

One of the first tasks of building this pipeline is to select and process an appropriate dataset to base the features and recommendations on. Following the creative process proposed by Muller [40], our system is primarily an influence generator that helps the creator better understand their problem from different angles. The selection of a corpus therefore decides the primary domain that the creator will be influenced by. Traditional word association algorithms typically look at how words are in a sentence, bigram, or trigram in order to relate words that have identical meaning or usage. To accurately identify synonymously used words these algorithms tend to use the largest datasets available, namely Google News, Wikipedia, or CommonCrawl. While detecting synonymous meaning and usage is valuable for tasks such as text classification, automatic summarization, or duplicate question detection, recommending

words due to synonymity is antithetical to creating the remote associations required for creativity enhancement.

Given our intention to help storytellers build secondary worlds, we selected a corpus that comprised primarily of poetry and novels. We used a dataset of 3000 public domain books from Project Gutenberg [32] consisting of letters, memoirs, novels, and poetry which span many different genres. Our Gutenberg dataset produced over 25,000 lemmas that could be recommended and over a million segments that collocated words. We explored other non-standard corpora, such as a database of book titles and articles processed from video game and tabletop role-playing game wikis. These smaller corpora tended to produce results that were either incredibly domain specific or completely random due to the limited number of collocations processed. We also explored the use of Wikipedia data, with the results described in Chapter 4. Of all of the options we examined, the Gutenberg dataset that we initially used produced recommendations that we considered the most interesting and creative for storytelling. Once the data is selected, we need to process it into segments that represent two words being associated.

Segmenting the documents differently will determine when two words are considered to be associated with one another. A large segment, such as a sentence or paragraph, will associate two words in a less obvious way than a small segment, such as a bigram. A bigram segmentation is likely to associate compound words, and to consider two words similar if their use is structurally similar. Our work requires that words are associated thematically, and thus we draw on literary conventions to select our segment size. We consider segmenting by sentence and by paragraph, as writers are typically encouraged to discuss the same topic at that scale. Other creative works, including Toivonen et al. chose to segment by sentence [20] after exploring bigram segments. For our work, we used an algorithm known as *TextTiling* [23] which uses topic analysis to separate the documents into multi-paragraph segments where each paragraph in a given segment has a high likelihood of having the same main topic. While we were able to use TextTiling for our work, we did find that the algorithm could take over 24 hours to segment our 3,000 documents compared to sentence segmentation which took less than an hour on the same workstation. While we used TextTiling for our evaluation, we also achieved good results using sentence segmentation; we describe both in Chapter 4. When segmenting, we also track the number of source documents that each term appeared in to be used later in the recommendation

process.

We select our corpus to produce influences that are directly inspired from literary works and our segment size to represent our desire for remote associations as opposed to obvious associations. As the segment size increases the resulting associations will become less relevant, though the reduced relevancy can lead to unexpectedly interesting results that are more likely to be a valuable, novel influence to the creator. After the documents are segmented the recommendation system can build a collocation matrix to describe the relationships between words; however, many of the most associated words are still be boring, obvious, and overused, so the system must first filter out any words that would be poor quality influences.

3.2.2 Filtering and Segment Reduction

The segments provide the set of possible associations for each word, though many of those words are not suitable as creative influences and need to be filtered out. In addition to removing poor quality words, the words that are not filtered are still represented by many different forms. Before reducing the segments to a bag-of-words representation the system must filter out as many poor quality words as possible and convert as many words into a common base lemma form as it can. This reduction not only strengthens the associations, as different forms of the word will still be associated with the same themes and less valuable words will have less influence, but also reduces the size of the matrix by an order of magnitude.

Our initial word filtering follows a four step process:

1. Remove stop words using a stop words table
2. Remove non-English words
3. Remove words that do not have an emotional rating in the DepecheMood emotional sentiment database
4. Remove words that appeared in our manually constructed blacklist of poor quality or offensive terms
5. Remove words tagged with part-of-speech tags that have been found to not contribute creative influences

The first step of our process is to check each word in the segment against a variety of blacklists. The first blacklist is a set of stop words that provide little information on their own, such as *the, is,* or *a*, for example. While some natural language processing tasks may make use of stop words to provide additional context in bigram analysis, the words are too common and provide little information for our system.

Next, the words are checked against a whitelist of U.K. and American English dictionaries. For our purposes, it is important to select a dictionary that does not contain proper nouns or made up terms that would be entirely irrelevant to someone writing for a secondary world, though this restriction may not be relevant to all applications. The dictionary whitelist mostly removes proper nouns, non-English terms, and non-alphabetical terms, but it is also useful for removing words with spelling mistakes which were included in the source material intentionally, such as in regional accents, or unintentionally. DepecheMood [51], a database that provides emotional scores for a word, also provides us a dictionary that we can use to ensure we only include words that have the features we require. While this does risk removing some otherwise valid terms from our corpus, DepecheMood's dictionary is very large compared to alternative emotional sentiment databases [49, 58] and we did not notice any loss in quality when using it, unlike the much smaller alternatives which would exclude many high quality recommendations.

The emotional scores provided by DepecheMood provide us with features that we can use to filter based on a threshold, rather than blacklists and whitelists. We use these emotional scores as a metric for how evocative a term is; section 3.2.3 describes the emotional features we use more in depth. While our results show that these emotional scores can be used to promote evocative terms to the top of recommendation lists, we reduce the scores down to simple sentiment values which may remove some valuable characteristics of the scores. Further exploration of these scores, as well as alternative models for emotional scoring of words, could provide more evocative results. It may be possible to integrate the emotional score as part of the *ranking* scheme, rather than a filter, and compare the emotional scores of the stimulus term to the recommendations to find interesting juxtapositions. While this work shows that emotional scores are a good indicator for how evocative a term is, there is still lots of room for exploration beyond this filter.

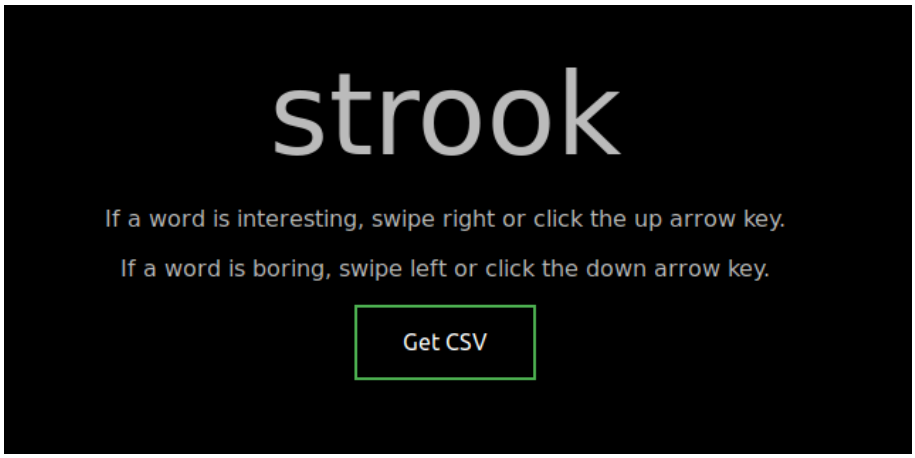


Figure 3.2: A tool built for the researchers to determine the part-of-speech tags that most frequently led to seemingly dull words.

The final word blacklist is a manually curated one; despite our attempts at automation, many formatting words, such as citations and footnotes, occurred in extremely irrelevant places. Words in the blacklist also included the most common words in our corpus, words that were considered offensive, or were otherwise of questionable quality. We found that many of the poor quality words we were blacklisting but were not deemed offensive or overly common were typically either an archaic term which were mostly removed by the DepecheMood dictionary, or were of the same part-of-speech. This curated blacklist and the terms that were not removed that seemed like they should have been automatically removed led to the development of our part-of-speech filter.

The part-of-speech (POS) filter considers how a word is used in a sentence, such as if it is a *noun* or *adjective*, and removes the word if the POS appears in our tag blacklist. The tool shown in Figure 3.2 was developed for the researchers to quickly analyze hundreds of words and look for tags that frequently represented poor quality words. The full list of tags that has been blacklisted is available in Appendix B. Many of the tagged words should have already been removed, for example punctuation and stop words, but the tags provide an additional way of guaranteeing less interesting words are removed.

Once words have been processed by the POS filter, English dictionary whitelist, and the blacklists, the remaining words must be reduced down into a common lemma

form. Converting terms into common lemmas allows us to easily recommend a common form of words, reduces our dataset size, and allows us to strengthen the relationships between words that appear together in different contexts. Many libraries offer tools to reduce a word into a lemma or to stem. While lemmas are typically formed by manually mapping words to their most common base form, stemming applies a set of rules to change a word into a common base form by looking at the last few letters. Stemming results would typically output words that were misspelled due to letters being removed, and thus we used a lemma dictionary. The lemmatizer provided by the spaCy natural language processing library helped to reduce most of the dataset, but some lexemes were still left without correct lemma forms.

Reducing the words into lemmas helped us to associate themes even when words are used in different contexts; for example, *flight* and *flying* may be used in different places, but we would still reasonably expect them to have common thematic collocations. While using lemmas helped avoid placing invalid words into our data, many lexemes were not automatically merged with the correct lemma. In addition to words not being merged, many words were automatically merged together that were spelled the same but had different meaning, for example the noun *ring* and the verb *ring*. For some tests, the part-of-speech was considered in the segment reduction and it was found to have more accurate relations; however, the more accurate relations seemed to do little to bolster results while increasing the total processing time and memory requirements beyond our 24 GB RAM capabilities. This could be solved by using an indexed file and slightly increasing the association lookup speed; however, any negative effect from some words being unintentionally merged (such as the noun *ring* and the verb *ring*) was minimal in practice.

Once each segment has been filtered and all of the lexemes have been converted into a base lemma, the segments can be represented as a reduced bag-of-words. During this reduction, the system also outputs a dictionary of 27,302 lemmas and their associated lexeme forms that can be used to parse user input. This reduced BoW representation makes it simple to build a collocation matrix and output any additional features that are valuable.

3.2.3 Collocation and Feature Extraction

The reduced BoW representation makes it easy to parse the data for collocations and various features that will help with removing any words that are of poor quality that

could not be removed by the filters. The resulting feature table contains an indexed list of N terms, where N is the total number of unique lexemes, and a sparse NxN collocation matrix. We selected a variety of features to calculate, including:

- The number of segments each word appeared in
- The total number of times a term appeared in the corpus
- The total number of source documents the term appeared in, calculated during segmentation
- The distribution of emotions that the word evokes
- The overall valence, or positivity, that the word evokes
- The maximum emotional score of a word
- The standard deviation of the emotions of a word

We use the *DepecheMood* database of words to calculate the emotional distribution of each word [51]. DepecheMood provides eight scores representing eight feelings: afraid, amused, angry, annoyed, don't care, happy, inspired, and sad. Each word in the list is also associated with a simple part of speech tag. The DepecheMood dataset was built by analyzing self-reported emotional responses to news articles and has proven to be both accurate and extensive with over 37,000 entries.

When selecting emotional scores for a word, we first try the part of speech that was most common in our analysis; if this is not available in the DepecheMood data, we move on to the next most common until a score is selected. We then reduce the emotional scores into three freatures: the maximum emotional score, the standard deviation, and the valence. The valence of a word was calculated by adding together the scores of *happy, amused,* and *inspired*, and subtracting the scores of *sad, angry*, and *afraid*. The *annoyed* and *don't care* attributes were not taken into account as they did not appear to meaningfully contribute to the valence calculation. This reduction provided us with simple values to use in our filters, but removes a lot of the nuance offered by having an emotional distribution. Future research could find additional features to consider in ranking or filtering that better represents interesting words and different emotional classifications may offer additional, valuable information; for example, emotional classifications which describe emotions as having opposites, such

as Plutchik's Wheel [45], may provide us with terms that appear frequently in two opposed contexts.

Once all of the features have been built, it is easy to build the collocation matrix. The collocation matrix M consists of N vectors $w = (w_i, w_j)$, where w_i and w_j are two different terms that collocated in the same segment. For each word w_i and w_j in segment $s \in S$, the value of $w(w_i, w_j)$ is increased by the number of times w_j occurs in s, denoted as $s(w_j)$ below:

$$w(w_i, w_j; w_i, w_j \in s) = \sum_{s \in S} s(w_j) \tag{3.1}$$

Some of the possible variations to calculating the collocations can reduce the impact of a paragraph that uses some words unusually often; for example, the BoW representation could set all values in the bag of words as one, as in the following:

$$w(w_i, w_j; w_i, w_j \in s) = \sum_{s \in S} 1 \tag{3.2}$$

or divide by the total number of terms $|s|$ in s to account for paragraphs that are unusually long, as in:

$$w(w_i, w_j; w_i, w_j \in s) = \sum_{s \in S} \frac{s(w_j)}{|s|} \tag{3.3}$$

Our system uses the method in Equation 3.1. While we did not deeply explore how each different collocation calculation method changed the resulting recommendations, we did find that the top recommendations for some stimuli were changed. A deeper analysis would be required to decide if one particular method performed consistently better than the others and how each collocation method interacts with the various association ranking schemes.

The resulting collocation matrix is largely sparse, with certain vectors being much denser than others. Once the complete collocation matrix is built, the data can be trimmed to Z values, 500 in our system, to decrease memory usage and processing time. For each word vector, the Z most collocated values remain unchanged while every other value is set to zero. This increases the sparsity of the matrix to improve memory and processing performance while having no major change on the accuracy of the vector based recommendations and actually seeming to improve the quality of recommendations, as described in the next section. While the choice to trim to

500 is subjective and could be tuned according to system requirements and personal preference, we were able to see a drop in relevancy with vector methods as Z reduced further and a decrease in relevancy as Z increased using weighting methods that promote infrequently used terms.

3.2.4 Single Word Recommendations

Once the input corpus has been reduced into collocation vectors the system needs rank associations from one word to another word. Simply looking at the highest collocated terms to a stimulus provides little information as the top collocated terms are extremely common throughout the dataset. To better rank words that are important to the stimulus we tested three different ranking systems: max-normalized TF-IDF, log-likelihood ratio (LLR), and collocation vector cosine distance. The latter two options use the vectors calculated in the previous section to segment similar words at runtime while the TF-IDF method involves pre-processing the data to obtain a new set of ranked vectors that require no runtime processing to rank.

The first method we tested was max-normalized TF-IDF as described in Manning et al.'s *Introduction to Information Retrieval* [34]. TF-IDF compares the number of times that a word appears in a document (term frequency, tf) versus the number of documents that the word appears in (document frequency, df). A high TF-IDF value indicates that the that a term appears very often in a document, that the term appears in very few documents, or both. This TF-IDF value therefore provides us with a combined measure of uniqueness and association.

By using maximum term frequency normalization, we can start to mitigate the influence of words that appear collocated across a large number of other words. To calculate the TF-IDF of a word w_i and another word w_j, we can look at the number of times w_j appeared alongside w_i, denoted as $tf(w_j, w_i)$, and divide by the maximum value in w_i, $tf_{max}(w_i)$ and dampen by a value between zero and one, a, which is noted by Manning et al. to typically be set to 0.4.

$$tf_{w_j,w_i} = a + (1-a)\frac{tf(w_i, w_j)}{tf_{max}(w_i)} \tag{3.4}$$

To build recommendations with TF-IDF the system pre-processes the collocation matrix and returns a new set of vectors containing the indices of collocated terms ranked by their TF-IDF values. When a user requests recommendations for a single

stimulus, the system only needs to return the weighted vector and the recommended words will be at the head of the list, in order. This pre-processing makes runtime recommendations as fast as a single table lookup and the sorted order means that most of the tail data can be removed as it will never be needed for recommendations.

Recommendations built with TF-IDF were often both thematically related and remote; however, the system often promoted words that were unique, but only seemed to be related to the stimulus by coincidence. To reduce these coincidental associations, we remove any terms which do not appear in at least some minimum number of documents. Removing these infrequent terms helps to keep terms related but also reduced the novelty of recommended words. This balance of filtering, trimming, and parameter tuning makes TF-IDF a bit unpredictable compared to the other vector methods. One problem that can occur with TF-IDF is when too many words are trimmed for having too few collocations; when this happens, the same words are often recommended for different, largely unrelated words. While we used TF-IDF due to its widespread use in natural language processing and importance ranking, Toivonen et. al [55] obtained good results in a similar association task using other statistical approaches.

Rather than ranking the importance of collocated terms, our second approach uses a likelihood ratio test to compare the use of two words in the corpus. While likelihood ratio tests are typically used to compare the fit of two different models, we can consider our stimulus vector our null hypothesis and compare each other vector to it as though they are observed experiments. By using log-likelihood ratio, defined as $L = -2log\lambda$ where λ is the likelihood ratio, we are able to account for the sparse nature of word collocations. LLR has been used effectively for tasks such as topic-focused document summarization [21] and bilingual word association [38]. This method works well for the sparse and contextually important data that our collocation vectors represent. To perform a recommendation with LLR, use the formula as described by Dunning [15], providing the necessary parameters from our collocation matrix and word features table obtained during the data reduction.

Early results using LLR to recommend terms tended to be similar to the results produced by TF-IDF with an average of 50% of the top ten recommendations for various stimuli being common between the two methods. With a lack of diversity between the methods and no apparent increase in quality, LLR was not further explored as an option and instead we focused on alternative methods.

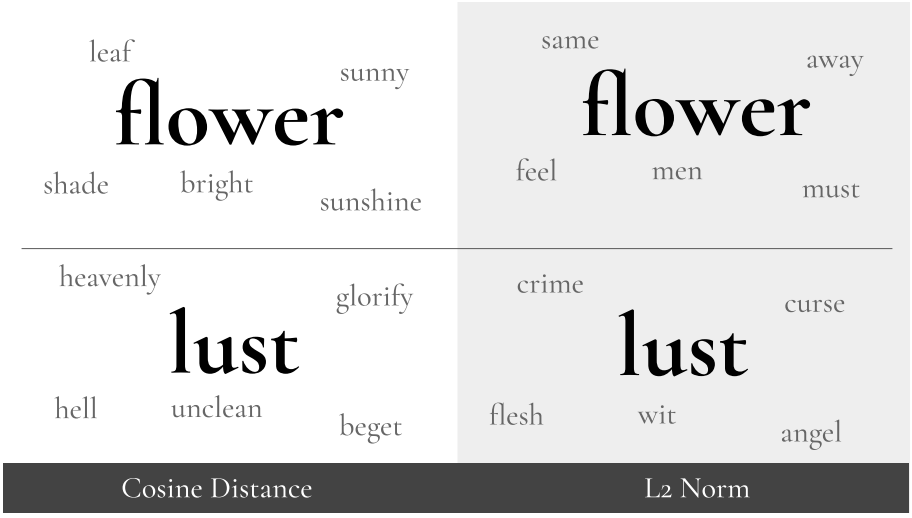


Figure 3.3: Comparing recommendations using L2 Norm vs Cosine Distance

The final method we considered treat the vector as a point in space and checks the distance between two points. For these tests, we used the trimmed collocation vectors and calculate the distances at runtime. We first tested using the L1 and L2 norms which each produced similar results to one another. The recommendations were not typically interesting and often not very specific to the stimuli, as can be seen in Figure 3.3. We then used the cosine distance using the same vectors, which was able to produce more interesting and consistently related results.

For our evaluations, we used the cosine distance to generate our lists due to the high level of diversity, remoteness, and overall quality of words. While the others may also provide interesting results, we would need to expand our user studies to better identify how the different terms that are recommended are able to evoke creativity and if there is a substantial difference between the quality of different lists.

Each of these three different methods produce some similar results to each other, with LLR and TF-IDF producing about 5 out of 10 words in common with each other and the cosine distance sharing about 1 out of 10 words with the other methods. While we selected the cosine distance for most of our testing due to its uniqueness compared to the other lists, a comprehensive test comparing the effects of segmentation and filters on the various ranking schemes would be required to adequately determine if any of the methods are optimal. Appendix A provides some examples of the words unique to each method for different stimuli.

3.2.5 Runtime Filters

The runtime filter makes use of the features that were built in 3.2.3 and can filter by:

- Minimum and maximum percentage of source documents the word appeared in
- A threshold for one of the emotional scores (maximum emotional value, absolute valence)
- WordNet relations

We performed automated tests to find which parameters seemed to have the most effect on the diversity of wordlists by generating lists with every permutation of parameters (where numerical values were appropriately quantized). The complexity of interactions between these filters does mean that they are difficult to analyze in isolation, but the tests still give us an initial basis to build adequate filters.

The first parameter is document frequency. When checking the document frequency, we could check against the number of segments that the word appeared in, the number of other words that collocate with the word, or the number of source documents that the word occurred in. Looking that the percentage of source documents provided a reasonable granularity to remove either overwhelmingly common or overwhelmingly obscure terms; however, during our automated testing we found that adjusting the maximum number of documents, independent of other filters, failed to make any significant change to the wordlists while adjustments to the minimum number of documents only became apparent at 5% and began to negatively influence wordlists beyond 10% by removing many remotely related terms and thus promoting more closely related ones. Where the document frequency filters begin to make an impact is when they are used in conjunction with more aggressive filtering which promotes more remote recommendations. For example, when recommendations are removed more aggressively using other filters with TF-IDF ranking, the system begins to promote words that appear in a larger number of documents. These filters help the system remove either entirely archaic or entirely generic terms that are more likely to occur when applying other filters. The maximum document filter will then begin removing more words than if the other filters were not applied so aggressively. It is unclear if this filter and our found thresholds would generalize across different datasets; however, it seems to reason that the same issues that the document filter solves would appear in alternative datasets and that some form of the document

frequency filter could be considered to overcome them.

The next parameters we considered involved the emotional scores of words. The three values we could look at were: the standard deviation between the eight emotional scores, the maximum score, and the absolute valence of the word. The standard deviation of the scores provided little change in diversity, though the maximum value and the valence both diversified the list and seemed to be directly impact the quality of words being produced. We suspect this is because the maximum value and absolute value of the valence represent the overall *arousal*, or evocative potential of the word. During our testing, we found that having a minimum threshold on either maximum score or valence introduced lots of additional diversity and was typically able to produce higher quality results with higher thresholds. Thresholds between 0.175 and 0.250 produced varied results, while thresholds above 0.250 were often too high and would reduce the candidate wordlist being reduced to a point that associations became irrelevant or entirely missing.

Finally, we looked into the relationships the stimulus had with the words in the list using WordNet [57]. With WordNet, we are able to find many different relationships between the stimulus word w_i and the candidate associated word w_j, namely whether the words are holonyms, meronyms, hypernyms, or hyponyms of each other. We suspected that words with these notable relationships were likely not remote enough from one another, and as such we can remove any words which are found to have these relationships. We found that removing these relationships did tend toward producing more remote wordlists; however, the effects were more subtle than we had first suspected.

For our evaluations, we used a dataset built using cosine distances, removing any words that appeared in fewer than 5% of the analyzed books, removed all obvious relations using WordNet, and used an absolute valence threshold of 0.175. While it is unclear if these are necessarily optimal parameters, they seemed to hold up well when the researches performed a blind test of various parameters.

3.3 Multi-Word Recommendations

Once recommendations are generated from a single word, we wanted to find a way to generate words from multiple stimuli to simulate creating a story in a rich environment populated by other themes. We wanted to represent a setting as a set of evocative

words that captured its themes and recommend new wordlists that represent new settings. These settings may be adjacent to the initial location, within the original location, or at the intersection of two different locations. Our work on recommending from a stimuli set is still early, but we were able to make some progress which enabled to us to explore the potential application space.

We worked with two different methods for multi-word recommendation, each only working with a small subset of the stimuli. We found very early on that it is difficult to meaningfully connect a large set of stimuli in a way that is clear to the user and working with a small subset of the stimuli makes recommendations seem more cohesive. One of the methods uses the nearby data for individual words while the other takes the mean of the collocation vectors and finds the closest words to that.

For each word we want to recommend for a given set of stimuli, our ranking method follows the following steps:

1. Choose a subset of k words, S_k, from the stimuli set S
2. Find the nearest words, N_k to each of the selected stimuli in S_k
3. Look at the intersection of the top m words from each set in N_k; if there is at least one available, return it, otherwise double m and repeat until finding a valid option or a maximum threshold is reached

Our vector distance method performs the following for each recommendation:

1. Choose a subset of k words, S_k, from the stimuli set S
2. Find the mean vector, v
3. Find the closest word to v and add it to the recommendations; if it is already there, randomly select from the top two, top four, top eight, etc., doubling the limit each time until either a valid word is selected or some upper distance threshold is reached

Once words are recommended, either through a single stimulus or multiple stimuli, the words can be run through another set of filters that can be tuned for how aggressively the system should remove potentially dull words.

3.4 Implementation

During the course of this project, many tools were developed as web applications to assist in evaluating different algorithms, parameters, and real world applicability. We created a swipe-based application to identify words that seemed dull at first glance. We used this to discover that some part-of-speech tags would consistently appear in poor quality words, leading to our blacklist seen in Appendix B. We also built an application to tag words in a word list according to our word quality indicators. We built an interactive map which allows the user to create new locations and generate wordlist recommendations by placing new points-of-interest down near existing locations to understand how the system might be used in the context of a worldbuilding tool. While these tools were built for the web, the core of the recommendation system was developed using Python 3.

The main libraries used in the data preparation pipeline include:

- **NLTK Tokenize**: The tokenize component of the Python Natural Language Toolkit exposes an implementation of the TextTiling algorithm
- **spaCy**: spaCy is a fast natural language processing library that can be easily run in parallel across large datasets and offers valuable information about words, including part of speech, lemma forms, and whether the word is a stop word.
- **TextBlob**: TextBlob is a library that is built upon NLTK and the *pattern* library and integrates with WordNet. TextBlob's *synset* functionality makes it easy to analyze the WordNet relations of each word for filtering. It also provides some of the features of spaCy; however, it performed much slower than spaCy in practice for tagging and lemmas.

Many other libraries were used at various times throughout development; however, these are the most useful and most specific libraries to this project. NLTK makes the implementation for segmentation trivial; however, it is important to note that running the TextTiling algorithm is a very slow process on large datasets. This is one of the major limiting factors for trying to use TextTiling on datasets such as Wikipedia or CommonCrawl. Once the data is segmented, the process of reducing and "lemmafying" segments can begin.

At this stage, it is valuable to start building a lemma dictionary that connects lexemes to lemmas. This makes it easier to quickly convert user input into a form that

```
nlp = spacy.load('en_core_web_sm', disable=['ner', 'parser'])
for segment in nlp.pipe(segments, batch_size=n_pipes,
                        n_threads=n_threads):
    lemmafy(filter_segment(segment))
```

Figure 3.4: Using a multithreaded spaCy implementation to build a lemma dictionary. The disabled functionality is not used and disabling those parsers greatly decreases runtime.

```
def _reduce(segment):
    words = []
    for token in segment:
        if token.lower_ in get_lemma:
            words.append(get_lemma[token.lower_])
    return ' '.join(words)

for segment in nlp.pipe(segments, batch_size=n_pipes,
                                  n_threads=n_threads):
    results.append(_reduce(segment))
```

Figure 3.5: Sample implementation of using spaCy with multithreading to reduce segments.

the system can understand and the values set can be used as a dictionary later on. Once the lemma dictionary is built, it can act as a whitelist for segment reduction. To find lemmas, there is a variety of libraries available. spaCy is able to quickly find tokens and lemmas for words and is able to easily be run with multiprocessing, as demonstrated in the implementation in Figure 3.4. The resulting lemma dictionary can be stored in any form, but this implementation uses JSON.

Once the lemmas are built, the segments can be reduced. To reduce the segments, the function can map any lexeme to its lowercase lemma form and remove any words that are not in the lemma dictionary keyset. A sample multithreaded implementation can be seen in Figure 3.5.

At this point, the segments can either be stored as is or converted into a bag-of-words using a *Counter* object from the *collections* library. This can also be done in the *_reduce* function if there is no more experimentation being done, such as testing out n-gram BoW representations. Building the counters from joined wordlists is a

quick operation so this implementation does not convert the representation at this stage so that the reduced segments can be used for additional experimentation that requires context preservation.

With the reduced segments, it is now useful to collect some feature information. If there is not already an indexed dictionary of words built from the lemma dictionary, it is valuable to save. This dictionary allows words to be referenced as integers and allows for some memory optimization through reduced data types provided by *numpy*, such as *uint16*.

Finding the right datastructure can be challenging for collocations. As the data is largely sparse, it may be suitable to use three lists to build a sparse matrix; however, the final implementation uses a variant with two lists, each of length N, that contain a list for each word. Each element w_i in the first list contains a list of indexes for each word w_j that collocates with w_i. The second list has the corresponding collocation frequency $tf(w_i, w_j)$. This representation makes it easy to quickly sort values using the numpy *argsort* function and trim values below a certain threshold at the expense of storage efficiency.

3.5 Evaluation Design

Evaluating the creativity enhancement provided by our system in an objective way is difficult without relying on user studies and the complexity of the system would require many studies to be run to adequately compare the quality of different methods. To avoid these lengthy studies, we developed a set of indicators which indicate poor quality words based on our core principles of creativity enhancement through word associations. For the purposes of our system, we are looking for words recommended for a stimulus that are:

- Related enough to connect the concepts in a meaningful way
- Remote enough to inspire new ideas, rather than simply reiterating the current ideas
- Novel enough that the user will not see the same theme repeated with different stimuli

We consider a wordlist to be practically successful if the wordlist helps overcome the blank page problem of writing stories better than alternative methods. In particular, we believe that our remote, yet associated terms can enhance the user's creative writing ability better than lists with close associations to the stimulus and completely random lists, where the words in the list are irrelevant to the stimulus. With these goals in mind, we developed the following indicators:

- **Unrelated** [**U**]: The word has no apparent relationship to the stimulus word.
- **Synonymous** [**S**]: The word provides little creative assistance as it is too synonymous with the stimulus.
- **Obvious Relation** [**O**]: The relationship to the stimulus is too obvious and does not stimulate ideas beyond the original meaning.
- **Dull** [**D**]: The word does not seem interesting or evocative on its own; it may be obscure, out of use, or lacks concreteness.
- **Repetitive** [**R**]: The word is very similar to one more many other words in the list.

It is important to note that these indicators, used individually, are not objective measures; individuals who are tagging words with these indicators may both agree a word is poor quality but tag it differently, or may disagree that a word is poor quality, and not flag a word at all. These indicators are better used to examine entire lists and identify how problematic a list is as a whole, as individual classifications may differ but the number of issues on a list will typically be quite similar. While they may not provide us with the ideal objective evaluation of worldlist quality, they do provide us with a metric that we can use to compare systems and can be further refined through user studies. More discussion on the inter-rater reliability on our internal tests with these indicators is available in Chapter 4.

Using these indicators, we developed an application that would pull data from various wordlists generated by different algorithms and the researchers could then select any words that had one or more issues, as seen in Figure 3.6. The researchers are presented with a stimulus and ten words and are asked to select the most obvious indicators that apply to each word.

Figure 3.6: Tool used by the researchers to evaluate words in wordlists.

While the individual indicators are subjective, we hoped that two researchers working with large numbers of wordlists would be able to find if there is a trend toward one method being preferred over another. These indicators can be used to develop heuristics to indicate when a particular system is likely to generate high quality lists; however, we require user testing to validate these indicators as an appropriate proxy value for detecting words that do not aid in creativity enhancement.

When selecting methods to compare to, we return to our goals. The two lists we chose to compare our results to were random, to ensure that our indicators held up against total irrelevancy, and *FastText* [6] which has been trained on the Google News corpus. We quickly noticed that evaluating purely random lists tended toward total irrelevancy and only included four lists, whereas FastText and our data, evaluated using the cosine distance ranking scheme, were tested against 29 distinct lists each.

While the indicators are able to highlight issues in wordlists, as we discuss in Chapter 4, user studies would be required to validate how our indicators translate to our practical goals of aiding creators with the blank page problem. If future user studies show that these heuristics are an appropriate indicator for creativity enhancement then we can use these indicators as a heuristic to rapidly evaluate system components.

3.5.1 User Study

In addition to our word quality indicators, we have organized an anonymized user study under CUREB-B with clearance #109529 in order to better understand how participants might perceive the various wordlists and whether the wordlists help to tell better stories. This user study has been designed to provide us with some initial

estimation of how well the different types of wordlists perform at aiding in user creativity and to help us understand how consistently users would rate wordlists with the indicators. While this study is not intended to completely validate or invalidate our word quality indicators, it should highlight any problematic components before expending the resources to perform multiple, targeted studies.

The user study consists of an online questionnaire that comprises of a demographic form, two warm up activities, and ten wordlists for the participant to read, use as inspiration, and rate using our quality indicators. After the participant has provided initial consent to collect their anonymized data, they are presented with the demographic form.

The demographic form asks the following information:

1. What is your age?
2. With which gender do you identify?
3. What is the highest level of school you have completed or the highest degree you have received?
4. Have you studied in what could be considered a creative field?
5. Have you worked in what could be considered a creative role?
6. In the past six months, how many fiction books have you read?
7. Do you have any experience playing tabletop role playing games (e.g. Dungeons and Dragons, World of Darkness, Pathfinder)?
8. If you said yes to the previous question, do you have experience running a campaign (e.g. dungeon master, game master, storyteller)?

The first three questions are intended to help understand the kinds of influences that the participant has when looking at associations within wordlists. We ask for the number of fiction books that the participant has read to try and have some estimation of their level of literacy and experience with written fictional stories; however, we recognize that this is only an estimate and is not an ideal indicator of literacy levels. We ask the remaining questions about creativity and experience with tabletop role-playing games to try and account for differences between people who are less

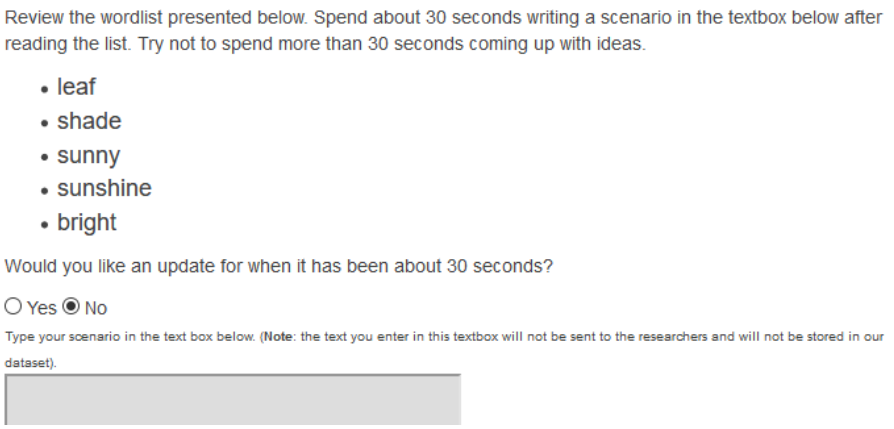


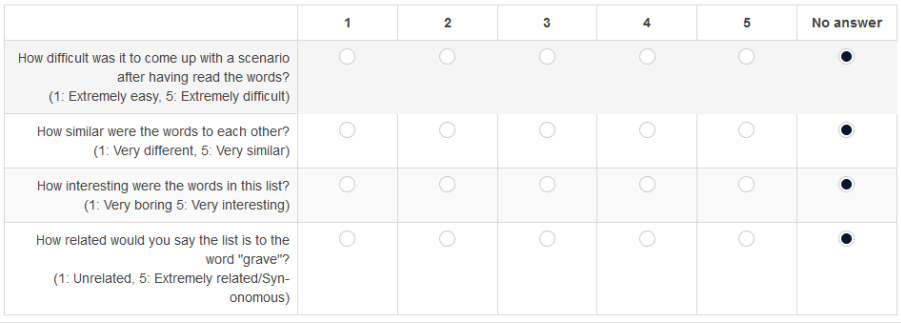
Figure 3.7: One of ten writing prompts given to the participant.

likely to practice creativity and improvisation and people who may regularly exercise creativity. Once they have completed the demographic form, the participant is presented with a warm up activity.

The warm up activity which is designed to prime the participant with how words can be used to represent settings. The participants are asked to read a passage from Charles Dickens's *Hard Times* and write a few words that come to mind after reading the passage. They are then asked to spend a minute coming up with a scenario that might take place in the setting described by Dickens. This priming is meant to help the participant see the wordlists as creative tools as if they were an intended user of our algorithm.

The participant is then provided with one more warm up activity. This warm up activity asks the participant to think of their favourite movie or book genre and to write about a scenario which might occur in this genre. This activity is intended to get the participant thinking creatively and warm up their writing of scenarios before being tasked with using our wordlists. They are then asked to find key themes of their scenario which were not explicitly written to try and get them to think about these words as themes rather than as explicit entities in the text.

Once the activity is done, the participant is shown ten stimulus words and their associated wordlist, four of which are from our algorithm, four from FastText, and two randomly generated. The final section tests the wordlist's ability to inspire creativity in the participants, to identify which qualities of the wordlists tend to help creativity more than others, and to measure the consistency of rankings between participants.



	1	2	3	4	5	No answer
How difficult was it to come up with a scenario after having read the words? (1: Extremely easy, 5: Extremely difficult)	○	○	○	○	○	●
How similar were the words to each other? (1: Very different, 5: Very similar)	○	○	○	○	○	●
How interesting were the words in this list? (1: Very boring 5: Very interesting)	○	○	○	○	○	●
How related would you say the list is to the word "grave"? (1: Unrelated, 5: Extremely related/Synonomous)	○	○	○	○	○	●

Figure 3.8: The four questions we ask after a participant has read stimuli wordlist recommendations and attempted to write a small scenario.

The participant is asked to review a wordlist and spend thirty seconds writing a scenario thank comes to mind after reading the list, as seen in Figure 3.7. They are then asked to rate various features of the experience and list on a scale of 1-5. The questions, seen in Figure they are asked are as follows:

1. How difficult was it to come up with a scenario after having read the words? (1: Extremely easy, 5: Extremely difficult)
2. How similar were the words to each other? (1: Very different, 5: Very similar)
3. How interesting were the words in this list? (1: Very boring 5: Very interesting)
4. How related would you say the list is to the word "flower"? (1: Unrelated, 5: Extremely related/Synonymous)

While the ratings the participant provides do not perfectly align with our quality indicators, they provide us with some insight into potentially problematic areas. We ask how similar the words are to each other to evaluate how repetitive the words in the wordlist are. We ask how interesting the words are to evaluate the *dull* indicator and combine synonymous words, obvious words, and unrelated words into a single rating when we ask how related the words are to the stimulus that generated them. By having the user rank between one and five, rather than binary decisions for each indicator, we hope to see a more nuanced understanding of the word quality indicators and reduce uncertainty when quality violations are not very severe.

We anticipate this as a pilot study for future, more targeted studies which can focus on a single component of our work. This study is intended to provide the initial

insight required to better understand how users respond to wordlist prompts for creativity and to identify any problematic indicators. After identifying any problematic indicators, we could benefit from two additional studies: one study which only asks participants to evaluate wordlists using our word quality indicators, either using binary decisions or with ratings; and a second study, which only asks users to make use of wordlists as creative prompts and rate how beneficial the different wordlists are.

Chapter 4

Results and Discussion

In this project we have explored methods to enhance the creativity of authors and world builders in games through the use of remote word association. In particular, our research has been in better understanding what properties make word associations more likely to enhance a user's creativity, how we can generate these associations, and examining how existing algorithms compare to our system. We have broken down the process of generating wordlists from user input into various system components, developed a set of indicators that help us to evaluate the efficacy of the wordlists in creative tasks, and performed some evaluation of the wordlists generated by various system configurations.

In this chapter, we discuss the exploratory work performed in the development of each system component and some of the interesting results that came up through our testing. We then briefly discuss some of the results of our wordlists generated by multiple stimuli and some of the challenges faced when moving from a single stimulus to multiple stimuli. Finally we compare our wordlists to alternative methods using our word quality indicators.

4.1 System Evaluation

This section discusses how the different components of our system perform under different configurations. Changes in each part of the system can have a big influence on the final recommendations and may interact with one another in unexpected ways; for example, one style of segmentation may perform better with one ranking scheme over another. Further analysis is required to understand how changes to different parts of the system will effect the outcome of each other component. This work

briefly compares different parts of the system to alternative implementations and discusses subjective qualities about the differences between them.

We begin by reviewing how we used a literary corpus to generate more literary results as opposed to larger datasets such as Wikipedia and compare wordlists generated under similar conditions with each. We will examine different segmentation techniques and compare results between our TextTile segments and sentence segments. We will then describe different methods that we used to relate words to each other and rank associations as well as how they performed under different conditions. We finish by discussing how the different parts of our filter performed and some of the challenges faced in building it.

4.1.1 Corpus Selection

When selecting a corpus for our problem we kept our literary and world building goal in mind by selecting a corpus that reflected these goals. The corpus represents the context for available external influences that we are providing the user. We theorized that popular fiction would be written in a way that naturally connected words in a more literary and thematic way than standard datasets, such as Wikipedia, and that those thematic associations would then be present in our generated wordlists more often than if we had chosen a large, non-fiction corpus.

An early example of a thematic association we found in our literary corpus was the word *fairy* and the word *flower*. While these two words are not necessarily semantically connected in an obvious way, they both tend to occur in similar contexts with one another in literature and therefore might be a creative influence in a writing task. These creative associations are not guaranteed from corpus selection, but instead a combination of corpus selection, remote segmentation to find remote associations, and selecting an association method that promotes novelty without removing relatability. The corpus selection is only the first, yet critical step in allowing these connections to be built.

To find out if selecting a literary corpus did produce results that were more likely to associate by theme than purely semantic meaning we generated wordlists with different stimuli using our literary corpus and generated a second set of lists using a subset of Wikipedia, as seen in Figure 4.1. We generated recommendations for each using the same parameters: sentence segmentation, cosine distance, and the same set of filter parameters. We chose sentence segmentation as the data parsed from

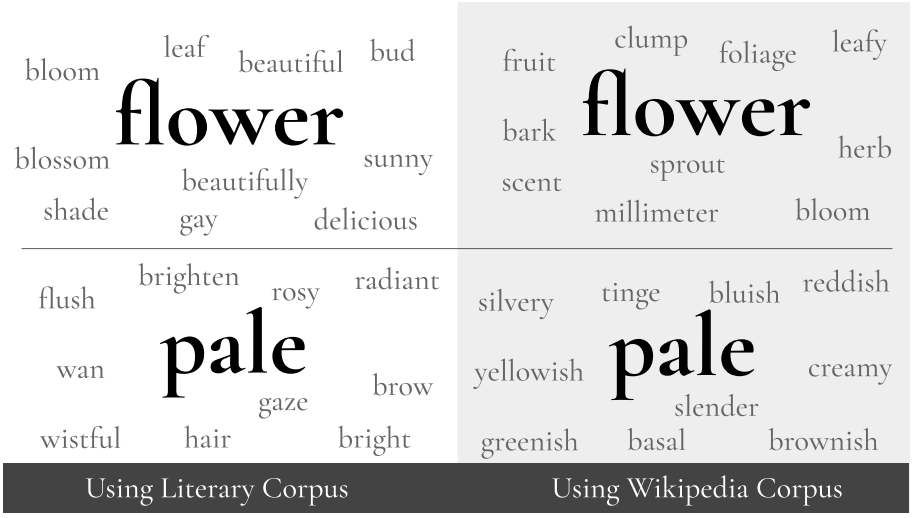


Figure 4.1: Comparing recommendations from using our literary corpora using sentence segmentation to a 20% random subset of Wikipedia using the same parameters. The highlighted words are stimuli, on the left are wordlists generated from our literary corpus and on the right are wordlists generated from the Wikipedia subset.

Wikipedia was already segmented by sentences and as we discuss in Section 4.1.2 both our TextTile segmentation and sentence segmentation provide good results. We used a 20% random subset of Wikipedia articles to reduce processing time which might effect the quality of associations; however, the selected stimuli are common terms and are likely to occur frequently in the data and the Wikipedia subset is still substantially larger than our literary corpus at around five times the number of segments.

In general, the different corpora can still produce good results. As we continued to explore filtering and ranking methods, we found that there were some words, such as the earlier *fairy* example, would only be associated with the Gutenberg corpus. These edge case associations are what we try to bolster through our various ranking and filtering methods and would be much more difficult to associate in the more literal and less literary corpora. It is possible that with new filters and ranking the alternative corpora could provide good results, but the Gutenberg dataset seems to readily find these thematic associations.

In addition to finding more literary associations, size of the corpus is a major factor in our tests. The complexity of system component interactions made it necessary that

we frequently build new data for our recommendations. Performing a full, sentence segmentation on Wikipedia with our current filters takes over three days just to find the whitelist and lemma mapping versus the two hours required for Gutenberg. This allows faster iteration and experimentation that helps to find new ways of bolstering the literary links.

4.1.2 Segmentation

In the creative process, the segmentation method represents the remoteness of our resulting associations. If the size of a segment is too small, such as a bigram, recommended words will be more likely to be directly associated to the stimuli and not provide a lot of additional influence than the stimuli alone. By extending this window to the scale of a sentence or paragraph we can rely on narrative rules to know that the words will still have some association, but will be less obviously related the larger the window size. For our purposes we tested the TextTiling segmentation, which associates terms by theme across one or more paragraphs, and also sentence segmentation, which associates words by at most one sentence.

To test TextTiling segmentation we tested our Gutenberg corpus on both sentence segmentation and TextTile segmentation. We generated ten words for each method and removed any that they had in common to compare the unique contributions of each method. The results some of these tests are available in Figure 4.2. These results show that both methods are capable of producing good results and would require more testing to fully evaluate.

The results are certainly influenced by the collocation ranking schemes that are used so it is important to analyze these recommendations through the lens of the creative process; in particular, our system's segmentation technique reflects the allowable obscurity of influencing themes. While random word association is typically used to help the creator make mental leaps between influences to possibly uncover new associations [7], we also recognize that we need to maintain some cohesion between our recommended themes and the existing themes of the world and story being written. We therefore position our recommendations somewhere between randomness and direct association.

For single word association we find the remote associations provided by TextTile are often able to promote creative connections as they border on randomness and

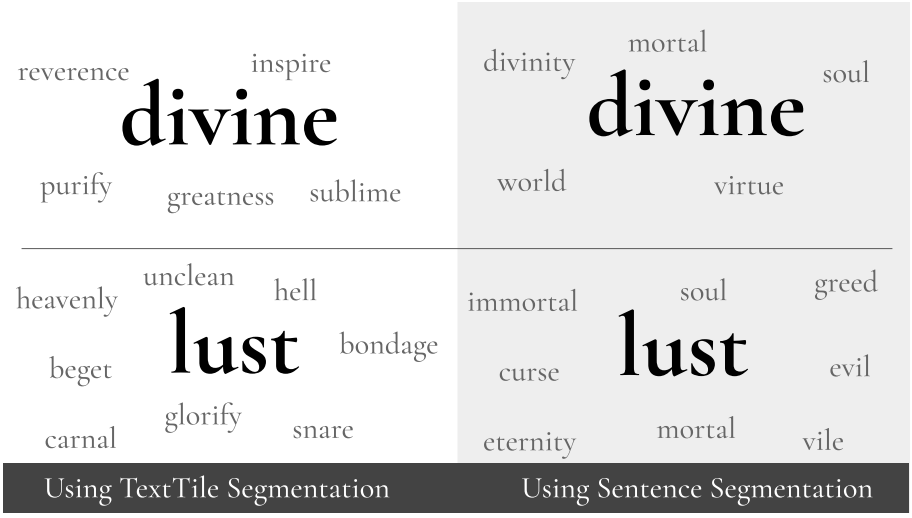


Figure 4.2: Comparing words that were unique to each segmentation method, comparing sentence segmentation to TextTile segmentation for two different stimuli. The highlighted words are stimuli. On the left are wordlists generated with TextTile segmentation and on the right are wordlists generated with sentence segmentation.

force the creator to work more to connect ideas. Looking at Figure 4.2, we can see that TextTile is more likely to produce some terms that more remote than sentence segmentation and that sentence segmentation still produces remote themes while being more generally associated to the stimuli. In our internal testing, we found that we generated more ideas when we were presented with a few very inspiring words rather than many words that were a bit more bland; as such, we focused our efforts on the remote associations of TextTiling and make use of aggressive ranking and filtering to emphasize the few, high quality associations and remove poor quality associations where possible.

Each ranking scheme provides different benefits: sentence segmentation will produce fewer irrelevant results at the cost of some very interesting connections, and TextTiling segmentation will produce very interesting associations at the cost of making the entire list feel less relevant to the input stimuli. Each ranking scheme can be used effectively but the remoteness of TextTiling allows us to better explore the edge case inspiring words that our system is ultimately trying to recommend more frequently. We use TextTiling going forward for our testing and try to use our ranking and filtering schemes to aggressively remove poor quality terms from the recommendations.

4.1.3 Collocation Ranking Schemes

Once words have been collocated according to the segmentation scheme, they can be associated by a variety of methods. Directly using the collocation data will typically produce the most *common* terms, not necessarily the most *relevant* terms. Measuring relevancy is difficult as terms can be relevant in different ways. Consider the word *flower*, for example:

- Lily is a hyponym of flower, as it is a specific type of flower
- Plant is a hypernym of flower, as flower is a specific type of plant
- Stem is a meronym of flower, as it is represents a part of the flower
- Blooming is a synonym of flowering, as it has the same meaning
- Weed could be paradigmatic to flower, as it could be used in many of the same places despite having different meaning
- Bee is related to flower by contiguity, as it often appears alongside flowers

While any of these relationships can prompt ideas we have found that contrasting and contiguous relations are more likely to spark novel yet related ideas. Early into this project we tried to discover some qualities of words which might help us find associations which evoke similar imagery while remaining remote enough to evoke new themes that can be used to create novel stories. While it is entirely subjective, we have found that related words tend to feel be more interesting and relevant to writing when they are purely contiguous or contrasting in nature and that words are typically less interesting when they are related by meaning the same thing, such as the hyponym, meronym, and synonymous relationships mentioned above. It is not a perfect indicator; there are some terms where even synonymous terms such as *bloom* can be interesting. To evaluate our collocation ranking schemes, we avoid looking at individual terms and instead focus on the overall list that was provided and try to categorize the relationships that are found.

We explore three different ranking schemes for this work: TF-IDF, log-liklihood ratio, and cosine similarity. TF-IDF associates words by their overall uniqueness in the corpus and their importance to the stimuli whereas the other two methods emphasize the similarity of the stimuli to other words.

Table 4.1: Sample of words recommended by our system when when distances are calculated using TF-IDF.

Stimulus Word	Generated List
flower	capsule, fertilise, pollen, bud, lily, stem, insect, blossom, bee, fairy
echo	footstep, cliff, hath, thunder, laughter, hollow, verse, poem, tremble, sin
grave	tomb, hath, dig, prayer, sorrow, poem, sin, bury, sad, bless
slaughter	massacre, unto, bloody, foe, slay, defeat, edition, hath, cattle, destruction
lust	holiness, righteousness, sinner, gospel, thine, tyrant, unto, cruelty, temptation, revenge
wander	poem, weary, moon, valley, snow, sad, grass, dwell, savage, darkness

The first method we were drawn to is TF-IDF. In theory, TF-IDF should find contiguous relationships as it exclusively looks at a words collocations and ranks within that. Table 4.1 shows a few of the wordlists generated by weighting the words by TF-IDF without any of our post-processing filters applied to the recommendations. TF-IDF was able to find many good contiguous and contrasting relationships, such as *flower* and *fairy*, *grave* and *sorrow*, and *lust* and *holiness*, it also produced many directly related results such as *grave* and *tomb* or *flower* and *stem*. There were very few unrelated terms; however, some quirks of the corpus and segmentation were made apparent with this method. One such quirk was the connection between *slaughter* and *edition*; these words appear to have no connection, though in the corpus the term slaughter appears alongside many footnotes that skew the influence of some irrelevant terms. These quirks can be fixed by either performing additional preprocessing of the corpus or by changing the segmentation implementation to ignore any meta text. TF-IDF was the most susceptible to these types of quirks and the alternative methods were not as influenced.

In addition to the relationships, we also had many words that simply failed to evoke much additional imagery. For example, while *fertilise* does not have a previously defined direct relationship to *flower* but does have a contiguous relationship, it fails to evoke any interesting imagery. Given that TF-IDF is capable of finding

Table 4.2: Sample of nearby words generated by FastText trained on the Google News corpus.

Stimulus Word	Generated List
echo	detune, stero, rasping, transfix, lightpen, pushbroom, rasp, spirant, digust, confabulate
flower	crocus, cornflower, peony, tulip, bouquet, garland, dewdrop, vase, leafage, bottlebrush
grave	sepulcher, sorrowful, catacomb, stone, cromlech, posthole, disquiet, crypt, blood-shed, drear
lust	greed, avarice, jealousy, debauchery, fornication, gluttony, madness, ugliness, cruelty, extravagance
slaughter	slaugher, carnage, massacre, killings, murder, dismemberment, decapitation, bloodletting, death, incineration
wander	drift, ruminate, leave, waylay, gather, espy, roust, hang, away, reck

many contiguous relationships, we compared the ranking scheme to recommendations generated by the FastText algorithm using an open source implementation by Martin O'Leary [41] as seen in Table 4.2. The FastText vectors were trained using the Google News corpus. The recommendations are often much more directly related, either producing words that are semantically related, such as *flower and cornflower*, or paradigmatic words that are used typically used in the same contexts but share different meaning, such as *lust and greed*.

Words generated by FastText were much more repetitive and were more directly semantically related; for example, words near *slaughter* tended to be very similar in meaning to the stimulus and to each other, such as *carnage, massacre,* and *murder*. When examining the list as a whole, there are many more words that fail to contribute in a meaningful way and far fewer contiguous relationships.

After seeing promising results when weighting the words using TF-IDF as it compared to FastText, we wanted to explore what other ranking schemes may also promote creative associations. The two main candidates we explored were log-likelihood ratio, which treats our collocation vectors as distributions and has been used by Toivenen et al. for creative associations [55], and cosine similarity, which has been used for many information retrieval tasks and vector similarity measures, including

Table 4.3: Sample of words recommended by our system when distances are calculated using log-likelihood ratio.

Stimulus Word	Generated List
flower	capsule, bud, pollen, fertilise, blossom, insect, bee, stem, bloom, purple
echo	footstep, cliff, laughter, moon, hollow, verse, murmur, noise, empty, tremble
grave	dig, tomb, prayer, evil, build, sin, song, faith, prove, bow
slaughter	massacre, bloody, destruction, defeat, victory, foe, kingdom, reign, slay, cattle
lust	holiness, tyrant, righteousness, vanity, gospel, cruelty, temptation, revenge, tempt, sinner
wander	weary, poem, search, journey, dwell, drink, soft, desert, memory, moon

sentence similarity detection [4]. Rather than looking exclusively for contiguity as we were with TF-IDF, these methods let us look directly for similarity and rely on filtering and segmentation for remoteness.

We generated two new sets of lists using the vectors representing the collocations to other words. We used the cosine distance between collocation vectors to generate the samples in Table 4.4 and we compared words as distributions as described by Toivonen et al. using the log-likelihood ratio [55], with some results of LLR generation available in Table 4.3. For these lists, we only looked at the top 500 most collocated words and set anything under this threshold to zero. This trimming helped to generate results quickly, taking us from over 26GB of RAM usage with our data structures and multiple minutes or processing per term to only 1GB of RAM and seconds to generate. Trimming the lists to the top 500 did not seem to degrade the list quality and even improved the quality of words when ranked using TF-IDF.

The results from our tests with LLR and cosine distance seemed to be more consistently related to the stimulus than those of TF-IDF. We compared the different methods to one another by generating ten recommendations for over 30 stimuli and removing any words that were common between the ranking methods, leaving only words unique to each method. While TF-IDF and LLR each produce an average of 5/10 unique words, using cosine similarity we obtain 9/10 words that do not appear

Table 4.4: Sample of words recommended by our system when distances are calculated using cosine distance.

Stimulus Word	Generated List
flower	leaf, shade, sunny, sunshine, bright, bloom, weed, delicious, shady, green
echo	ear, sound, breathe, loud, stir, strain, breath, hush, faint, star
grave	sad, solemn, careless, cast, bitter, fair, seek, alive, depart, blind
slaughter	bloody, rout, battle, slay, victory, foe, combat, flee, fight, victorious
lust	eternal, everlasting, heavenly, unclean, hell, glorify, beget, pollute, sinful, bondage
wander	lonely, stray, linger, haunt, restless, solitary, wand, gloomy, grow, weary

using any other method. The results of these comparisons is available in Appendix A. The consistent relationships of the vector methods combined with the novelty of the cosine distance made cosine distance the optimal choice for our continued testing.

While comparing the methods, we also wanted to check if terms that appeared across multiple, similar methods were typically of higher quality than the words that were unique to a single method. Table 4.5 shows a comparison of words that are common across at least two methods or unique to TF-IDF or LLR; however, unique results from cosine similarity were not included in the table for brevity due to the large number of unique terms. To compare the unique terms from cosine distance, it is easier to compare all of the results generated as seen in Appendix A.

The results in Table 4.5 show that words common to the systems are consistently closer related to the stimulus, though they many not necessarily be more evocative. These results could be expanded in the future and ideally a user study would help to quantify any evocative loss between looking at commonly suggested words and words which are unique to individual systems. The intersection of multiple different ranking schemes does seem to be more appropriate when thematic consistency is of more concern than finding unlikely evocative influences. It is possible that the optimal approach is to use multiple ranking systems in conjunction with one another to remove irrelevant recommendations and promote remote contiguous relationships but further exploration is required to better understand how these systems can work together.

Table 4.5: Sample of words that were common between the top 10 recommendations of at least two ranking schemes or unique to a single method.

Similarity	Stimulus	Recommended Words Common to Multiple Methods
common	flower	bloom, lovely, fairy, seed, bee, blossom
unique	flower	grass, kiss, poem, valley, fruit, joy, song, wonderful
common	echo	noise, tremble, laughter, moon, verse, murmur
unique	echo	distant, shade, sigh, tale, gun, poem, roar, valley
common	grave	bury, song, poem, sorrow, tale
unique	grave	ago, beneath, drink, evil, teach, bless, gate, gaze, poet, soldier
common	lust	throne, temptation, tempt, hell, fame, cruelty, wick, gospel
unique	lust	base, crime, angel, flee
common	wander	poem, journey, moon
unique	wander	desert, dream, drink, grave, leaf, pleasant, wave, gate, grass, music, poet, savage, song, valley

While the ranking scheme is one of the most important components of our system, we also built multiple filters to find ways of automatically removing poor quality words from recommendations. These filters bolster the final recommendations by removing inherently poor quality words and words with poor quality relationships to the stimulus.

4.1.4 Filtering

To bolster the recommendations generated after the collocations have been ranked we apply a variety of filters to the words to remove any terms that are of poor quality. In our system there are two points of filtering: the initial blacklist filter, which removes terms based on simple dictionary blacklists before collocations are calculated; and the feature blacklist, that is run after collocations are generated to remove words based off of features that are calculated after the bag-of-words formation. While the feature blacklist is capable of being during the pre-process phase along with the blacklists, it is used at the end of our system so that the parameters can be tuned according to the user's preferences.

The blacklist filters are applied to remove any words that have been found to represent poor qualities for word recommendations at a basic level by removing words based on basic word properties, for example, words might be removed if they are: a proper noun, a non-English word, a stop word like *is* or *the*, or if it is one of the most common words in the English language. While the blacklist filters could be applied during the recommendation stage, they are most effective before building the dataset to reduce the number of tokens in the dataset and optimize the system and do not have any additional parameters that need tweaked. When the system is reducing segments into their bag-of-words representation the words are run through the blacklist filter and any words removed at this stage will not appear in or influence any collocation vector.

The second set of filters we use is the feature filter. The feature filter uses different features of the word, such as emotional rankings and document frequency, to remove words that might not be deemed novel or interesting. The threshold for each feature factored into the filter is adjustable and allows the user to determine how aggressively the system should filter poor quality words by each feature. The words that are filtered out at this stage can still influence the vector based ranking schemes; for example, removing the term *brown* because it has a low emotional magnitude would make it so that it will not show up in recommendations but the term still occurs in other vectors and will be used in distance calculations and likelihood ratio tests. While this filter can be used at the segment reduction stage along with the blacklist filter the ability to adjust how aggressively terms are filtered is valuable for experimentation and user customization. When the filter is applied during the recommendation phase, the system will:

- Rank all candidate words by how associated they are to the stimulus
- For each candidate until 10 candidates are selected:
  - Lookup the candidate features
  - Test the features against the filters
  - If the candidate is valid, select the word; otherwise, discard the term and continue searching

The primary goal of the blacklist filter is to remove terms that unnecessarily increase the amount of data we need to store and process without negatively impacting

the quality of our recommendations. Unfiltered, our corpus contains 630,558 unique tokens including character names, locations, and non-English text. During the segment reduction phase we were able to remove 577,646 tokens, leaving us with 52,912 lexemes that were reduced to 27,302 total unique lemmas. The most common issues that we found, in no particular order, were:

- Tokens with hyphens in them
- Spelling mistakes
- Proper nouns
- Made up words
- Stop words
- Numbers

Table 4.6 shows some data collected when we ran the FastText algorithm trained on Google News data through our filter to obtain a list of 10 filtered recommendations. Without filtering, these terms would have been recommended to a user as the closest association. The removed words in Table 4.6 are clearly of poor quality and tend to be jarring when they are recommended. During our evaluations we were frequently distracted from our goal by spelling mistakes such as *digust* and hyphen mistakes such as *job–and*. These extra terms take up unnecessary space in the data structure and add noise to the vectors. When spelling mistakes are not removed from the data they are represented as infrequent tokens and their infrequency will lead to them being seen as important features. This accidental bolstering can lead to words being connected by a spelling mistake by fluke or can increase the likelihood of spelling mistakes being recommended to the user, even when the correct word is otherwise common or unrelated.

We generated sets of ten recommendations for 33 different stimuli using the FastText algorithm. To obtain ten filtered recommendations we needed to generate an average total of 28.18 words per stimuli (or an average of 18.18 words being filtered out). Some stimuli required dozens of terms to be filtered, such as *score* which required 118 total recommendations before recommending ten words that were not blacklisted. Other stimuli required comparatively little filtering; for example, the stimuli *flower,*

Table 4.6: Words generated by FastText and run through our filter. Shown: The stimulus used to generate the wordlist, the number of unacceptable words removed before finding ten acceptable words, and a sample of words considered unacceptable.

Stimulus	n Removed	Sample Removed Words (at most 5 shown)
echo	16	detune, stero, pushbroom, spirant, digust
grave	12	posthole, whinstone, drear, unresting, perpetuall
parade	19	camp-out, wedding-themed, tickertape, face-painting, knees-up
spirit	13	self-will, good-neighborliness, open-mindedness, spirit-, heartedness
score	108	pointscorer, career-low, minus-2, team-best, 90-point
job	57	dayjob, pre-job, single-task, job–and, jobholder

lust, wander, slaughter, ardour, and *pale* all only required three or fewer words to be filtered out before obtaining ten valid recommendations. Additional data regarding our FastText filtering can be found in Appendix D.

Once the blacklist filter has removed most of the obviously poor quality terms from our corpus we tested our feature filter on the remaining terms. We tested various configurations of thresholds and features that we had available and discovered that the feature filters were able to better emphasize evocative and rare terms. Table 4.7 shows some of the results of our filtering with different stimuli when only one feature filter is applied at a time.

One novel filter we used is the emotional ratings for the recommended words. Using the DepecheMood dataset [51] we were able to find an emotional ranking for each word and use this to represent how evocative a word might be. We reduced the different emotional scores into two valuable features: the absolute valence, representing the overall positivity or negativity of the word; and the max emotional score, representing the highest emotional rating, or emotional intensity, of a word. These emotional features often successfully promoted terms that were more evocative and interesting, in particular when we made use of the maximum emotional score. While aggressive thresholds like those used in Table 4.7 can remove high quality words such as *depart* and *solemn* they also tend to promote high quality words in their place.

Table 4.7: Words added and removed from a set of unfiltered, cosine ranked recommendations when one feature filter is applied; wordlists presented are truncated to at most three terms.

Applied Filter	Stimulus	Added After Filter	Removed With Filter
Valence: 0.25	flower	airy, golden, shape	sunny, shady, shade
Valence: 0.25	echo	stifle, smother, babble	stir, loud, faint
Valence: 0.25	grave	proud, shrink, marvel,	depart, sad, alive
Max Docs: 80%	flower	golden, lovely, blossom	leaf, green, bright
Max Docs: 80%	echo	sullen, murmur, burst	star, breath, sound
Max Docs: 80%	grave	forever, fate, youth	alive, seek, cast
Emotional Max: 0.25	flower	wither, blossom, airy	sunny, shady, green
Emotional Max: 0.25	echo	stifle, groan, smother	stir, loud, faint
Emotional Max: 0.25	grave	proud, melancholy, shrink	depart, sad, solemn

When filtering is too aggressive some terms can be left with very few remaining associations. When using vector based ranking schemes this leads to recommendations that are either very archaic or unrelated to the stimulus. The minimum document filter tends to do very little on its own, but it can be used with other filters to remove the archaic terms promoted by aggressive filtering. Unrelated terms will be further promoted and our system has no current way to detect when terms are too remote to recommend. While unrelated terms are problematic for recommendations, we also had to build a filter to detect when recommendeations were too obviously related to the stimulus.

While most of the filters we used examine the recommended word in isolation, we made use of word relationships defined with WordNet [57] to remove recommendations based on their relationship to the stimuli. We looked at each recommendation and checked how WordNet classified its relationship with the stimulus. To try and promote interesting relationships we would remove any recommendations that were a synonym, holonym, or hypernym of the stimulus. The words themselves may be of high quality but may not contribute any novelty to the wordlist due to the direct and likely obvious relationship to the stimulus. This filter did not remove many terms, but the terms that did get filtered were always closely related, such as removing *newspaper* as a

Table 4.8: Example data used as stimuli for our mapping prototype to explore the application of multi-stimuli associations.

Location Name	Representative Stimuli
Stark	ancient, honour, defend, protect
Bolton	blade, flay, torture, insidious, betray, hate, sickly
Tyrell	gentle, serene, flower, wealth, conspiracy

recommendation for *paper* and *passion* from being recommended for *ardour*. While *passion* is usually a high quality recommendation it may not serve well as a creative influence as it is synonymous with the stimulus *ardour*.

The inclusion of our emotional filters and our emphasis on aggressive filtering increased the overall quality of the recommended wordlists. We found that emotional scores are a good measure of how evocative a word is and were able to use this to promote more evocative influences to the user. Combining these emotional scores with basic frequency filters and directly targeting any obvious relationships through WordNet led to higher quality lists at the cost of losing some potentially high quality recommendations. By applying the simple blacklist filter during segment reduction and applying the feature based filters at runtime, the user is able to choose how aggressively the system removes words and adapt it to their problem.

4.2 Multiple Stimuli Recommendations

Most of the research for this thesis was focused on recommending evocative words for a single stimulus; however, our early prototypes demonstrating the use of word recommendations to represent themes on a map showed a strong case for making use of multiple stimuli to represent the many themes that make up a rich fictional setting. In our prototype we represented different locations as a set of themes, as demonstrated in Table 4.8, and each location was given a position on the map. Our goal is to allow users to place new locations on the map and be presented with suggested themes based on the themes of nearby locations.

The stimuli used for different locations do not necessarily need to be user generated. To test our multi-stimuli recommendations, we generated a new set of locations

Table 4.9: Words extracted from various novels for multi-stimuli recommendation.

Book	Sampled Keywords
Dracula	monster, peril, sleep, blood, fear
Frankenstein	ardour, mockery, monster, overwhelm, destruction, creature
Wonderful Wizard of Oz	rainstorm, genie, emerald, witch, charm, castle

based on popular books and tested different methods to see if they seemed to align with other themes in the novel. To extract these themes, we ran a max-normalized TF-IDF on Bram Stoker's *Dracula*, Mary Shelley's *Frankenstein*, and L. Frank Baum's *The Wonderful Wizard of Oz*, using the IDF values provided by our literary corpus. From these a few standout terms were manually selected and can be seen in Table 4.9. These terms were then used to generate new recommendations which could act as themes for new stories within the book.

We developed two methods to generate recommendations from multiple stimuli using what we had learned from single stimulus recommendations: a vector based method that treats the stimuli as a single vector, and a method which looks for words that are associated to each stimuli when weighted by TF-IDF. The vector based method assumes that words near the mean vector of multiple stimuli, calculated using the cosine distance, will represent themes that are somewhere between the stimuli. Our intersection method is based on the assumption that words that are important to multiple stimuli can represent novel connections between the themes.

Very early into our prototype testing we learned that trying to find recommendations that relate to all of the provided stimuli tended to produce results that felt unrelated and boring. The task of relating multiple words by a single association is similar to that of the remote associations test (RAT) [35] which has been used to quantify the creativity in humans and is proven to be a difficult challenge for most people. Given that the RAT is a challenge for humans to perform we designed the system to use only a subset of the provided stimuli so that the recommendations could still be readily associated to the stimuli by the user.

To test each of our methods using the themes generated by our novels we used random subsets of the stimuli ranging from 1-3 stimuli selected, and we tested with all stimuli to test our theory that the subsets are required. The recommendation lists

change between each run due to the stochastic selection of stimuli and the similarity of the stimuli will determine how diverse the resulting recommendations will be. For each word that is recommended the system attempts to selected a new, random subset of stimuli using the stimuli seen in Table 4.9.

The results of our vector based recommendations, seen in Table 4.10, show that using all of the stimuli does result in recommendations that are more generic and common in the dataset. The results for using all stimuli for *Dracula* and *Oz* recommend words like *till*, *full*, and *fine*, while using fewer stimuli produced results that were more evocative. Using a single stimulus produces words that tend to be distinctly related to a single stimulus and do not seem as cohesive as the lists with two or three, though the quality of the recommended terms is higher. The increased quality of recommendations from using a single random stimuli seems to outperform the lesser quality, yet more cohesive lists recommended by multiple stimuli. As the number of stimuli used to generate recommendations increases the recommendations become too remote for our creative purposes.

When we tested our recommendations using TF-IDF we attempted to generate ten recommendations for each book by looking at the top intersecting words of random 1-3 word subsets of the stimuli. The system selects a random subset of stimuli and builds a candidate set from the intersection of the top two words. If no words intersected, the system would begin intersecting the top four, then top eight, doubling each time until it surpassed the upper limit set by the user, in our case fifty words. Beyond fifty the system began to fill out all of the recommendations consistently at the cost of irrelevant words being added to the list. If a user chooses that populating a list is more important that obtaining relevant recommendations they can increase the upper limit incrementally after exhausting all candidates below it. To better represent the recommendation system we use a low limit of fifty in Table 4.12, but Table 4.11 shows some of the poor quality results obtained when intersecting all of the stimuli for each book when incrementally increasing the upper limit to five hundred.

In contrast to the results obtained by vector based method, the results in Table 4.12 show strong recommendations with multiple stimuli. The themes are interesting and the addition of extra stimuli seems to take the wordlists in a different direction than the single stimuli recommendations which, similar to the vector method, are mostly closely related to a single stimuli from the provided list. Some of the words

Table 4.10: Multi-stimuli recommendations when using the mean collocation vector and searching for words with the smallest cosine distance to the mean vector using the themes in Table 4.9.

Book	n Stimuli	Recommended Words
Dracula	All	terrible, rest, awful, blind, till, break, save, lose, dare, dead
Dracula	1	danger, sleepy, monstrous, knowing, asleep, save, warn, crush, dare, frightful
Dracula	2	bold, stir, alive, till, save, bind, lose, struggle, crush, dare
Dracula	3	stir, terrible, break, cold, save, warn, crush, dare, dead, grave
Frankenstein	All	utterly, strong, weak, feeble, full, monstrous, bear, bind, seek, crush
Frankenstein	1	passion, ardent, wonderful, perish, bear, ugly, fatal, destroy, fancy, grave
Frankenstein	2	body, weak, wonderful, bear, blind, sustain, touch, destroy, strength, cease
Frankenstein	3	strong, weak, destroy, wonderful, careless, bear, seek, touch, fancy, cease
Wonderful Wizard of Oz	All	fashion, pleasant, splendid, full, favourite, fine, bear, fair, delight, grand
Wonderful Wizard of Oz	1	rain, magician, fetch, magic, golden, attendant, drench, admire, shower, dead
Wonderful Wizard of Oz	2	don, modest, guard, join, admire, delight, golden, attendant, beautiful, delightful
Wonderful Wizard of Oz	3	don, favourite, magic, delight, admire, join, attendant, beautiful, delightful, fancy

Table 4.11: Multi-stimuli recommendations when taking the intersection of TF-IDF weightings using *all* themes in Table 4.9. The upper limit of associated terms was incrementally increased up to 500 to obtain ten results for each book.

Book	Recommended Words
Dracula	beat, fit, joy, heavy, ear, heaven, rock, self, burn, dream
Frankenstein	except, noble, forth, learn, produce, self, dream, dead, town, art
Wonderful Wizard of Oz	except, evening, strike, men, forth, sing, wind, mind, town, away

from the intersecting stimuli, namely *scarecrow, ruby,* and *wizard* as they relate to the Wonderful Wizard of Oz stimuli seem to derive directly from the source material. This suggests that the system is successfully finding strong thematic links between the otherwise unrelated terms. The stimuli provided for Oz are the most diverse of the other stimuli and are less likely to appear in combination in other texts. This can account for the intersecting recommendations being so close to the source material. This diversity also led to Oz being the only source material that was unable to provide ten recommendations for three stimuli.

We were able to show that using many stimuli when recommending new words leads to poor quality, unrelated recommendations. Our system is able to produce higher quality recommendations by selecting small, random subsets of the stimuli to generate recommendations rather than the entire set. While the exploration into using multiple stimuli to generate recommendations is early, we were able to show some of the problems that occur when translating our single word recommendations to multiple stimuli, provided a method which intersects TF-IDF weighted remote associations to produce promising recommendations from multiple stimuli that are more diverse than only using a single stimuli, and present these findings as the starting point for future research.

4.3 Evaluating the Quality of Recommendations

Our system is intended to aid writers and game developers when developing new worlds and telling new stories. We would require user studies to evaluate how well

Table 4.12: Multi-stimuli recommendations when taking the intersection of TF-IDF weightings using the themes in Table 4.9. Ten recommendations were attempted, but some were unable to find enough recommendations when only looking at the top 50 words.

Book	n Stimuli	Recommended Words
Dracula	1	asleep, awake, terror, bless, dread, curse, slumber, imminent, flesh, dragon
Dracula	2	beast, hero, joy, terror, behold, soldier, army, beneath, passion, sail
Dracula	3	beat, dog, queen, creature, joy, terror, guard, army, passion, beneath
Frankenstein	1	wound, beast, terror, canto, verse, poetry, emotion, hell, dragon, grief
Frankenstein	2	beast, poem, hero, terror, leaf, storm, passion, savage, island, charm
Frankenstein	3	rush, evil, poem, joy, shape, terror, wave, passion, island, behold
Wonderful Wizard of Oz	1	garrison, poem, gusty, fatty, baron, downpour, delightful, sultan, earl, scarecrow
Wonderful Wizard of Oz	2	tale, poet, fairy, diamond, poem, ruby, lovely, song, scarecrow, wizard
Wonderful Wizard of Oz	3	tale, fairy, poem

our system is able to help actual users bypass the blank page problem. In the absence of this, we have developed some proxy metrics that we can use to evaluate parts of the system. While we have designed a user study to help evaluate the performance of our metrics, this section describes how we have started to make use of certain indicators to compare the qualities of our recommendations to alternative methods.

The indicators we selected were based on our own usage with the recommendations and typically indicate words that do not bolster creativity. We evaluate wordlists using our indicators by looking at a wordlist and the stimulus that generated it and selecting the most obvious issue with any words in the list from the following set of indicators:

- The word has no relationship to the stimulus
- The word is synonymous to the stimulus
- The word has too obvious a connection to the stimulus that it fails to stimulate new ideas
- The word is dull
- The word is very similar to one more more other words in the list
- The word has other issues, typically spelling mistakes or archaic terms

4.3.1 Comparing the Wordlists

To test our indicators, we examined 62 unique sets of recommendations each, 29 of which were generated by FastText, 29 were generated using our cosine distance ranking scheme and somewhat aggressive filtering, and the remaining four wordlists were generated at random. The indicators are subjective and individual selections can vary. We intend for these indicators to highlight patterns of issues in algorithms rather than necessarily individual words or lists. By counting how many words were flagged for various wordlists we hope to identify what qualities are present in each recommendation system and eventually test if certain qualities in wordlists can aid in creativity better than others.

Table 4.13 shows some of the results of our indicator tests when testing for random and the full results are available in Appendix E. In our test of 29 different stimuli our system outperformed FastText in all but one stimulus, *paper*, where both FastText

Table 4.13: Average number of words out of ten which were flagged as poor quality for stimuli which were tested with random.

Stimulus	**Ours**	FastText	Random
echo	**2.5**	8	9
flower	**5**	7.5	6.5
lust	**2**	5.5	7.5
slaughter	**4**	8.5	7

and our system flagged nine problematic words out of ten. Early into our tests we theorized that words that were not very evocative would be less likely to associate with evocative words and that the distance from manually curated evocative words may themselves be able to be used as a possible metric for how interesting a word could be.

Looking at the words flagged with the most issues we have: *paper, card, wing, majority,* and *pocket*; conversely, the words with the fewest issues were: *divine, ardour, creation, wander,* and *lust*. The words that generated recommendations with fewer issues are clearly more evocative for storytelling than the stimuli that generated poor quality recommendations. This seems promising enough to continue research into methods that could use these manually curated evocative terms as "anchor points" that can be used to influence the ranking of other recommendations based on their association with known high quality terms.

To further examine the quality of words in the systems we calculated the average number of words flagged with any indicator, with the results made available in Table 4.14. When ten recommendations are generated, we see that on average FastText words are flagged 83.3% of the time, random words are flagged 75.0% of the time, and our method is flagged only 38.1% of the time. If we assume that our indicators do work as a metric for terms that are problematic, we see that FastText is only capable of producing 1-2 valuable words compared to the average of six that our system produces. Given these metrics we see that our system performs consistently better than the alternative systems that we tested, but we further explored what issues were being flagged most often with each system.

Before our testing, we assumed that FastText, being based on non-fiction works

Table 4.14: Comparison of word number of words flagged as poor quality, including on average how many more words were flagged on a per-stimulus basis

Method	Mean	Median	Stdev
Ours	**3.81**	**3.5**	**2.07**
FastText	8.33	8.5	1.47
Random	7.50	7.25	1.08

Table 4.15: Comparison of different issues occurring in each system tested.

System	Unrelated	Synonymous	Obvious	Dull	Repeating	Other
Ours	**42.75%**	**3.05%**	**1.53%**	**35.50%**	**13.36%**	**3.82%**
FastText	4.33%	18.94%	14.93%	20.71%	31.78%	9.31%
Random	69.33%	0.00%	0.00%	22.67%	0.00%	8.00%

and built for traditional word embedding problems, would likely find relationships that were very obviously related to the stimulus and through this produce many repetitive recommendations. We suspected that random should produce recommendation that are unrelated to the stimulus and that our own system was most likely to produce unrelated recommendations. Table 4.15 shows the percentage of issues that were flagged for each different system. The data does not reflect every issue that occurred but instead the most obvious issues with each word. These results show that the issues we suspected would occur in the random and FastText tests were the predominant issues, with the main issue of FastText being repeating words at 31.78% and high percentages of both obvious and synonymous recommendations. The random system performed as expected with 69.33% of issues being flagged as unrelated to the stimulus. In our own system, 42.75% of the flagged issues were flagged as unrelated and 35% were flagged as dull. We very rarely had terms occur that were synonymous or obviously related.

Our results with the indicators confirmed our original theories and shows that, if our indicators are a good proxy metric for improving a user's creativity, that our purpose-built system will outperform alternative solutions in user tests. While our results seem to correlate with our original theories of how the systems would perform,

we next needed to examine how consistently we agreed upon tags and agreed upon whether a word is poor quality or not.

4.3.2 Agreement Between Quality Ratings

To compare our agreement we make use of the simple matching coefficient (SMC). The SMC measures the similarity between two binary sets where both *1* values and *0* values contain symmetric information. We compare our agreement at three levels:

- Stimulus by stimulus, per system
- Overall agreement per system
- Overall agreement between the systems combined

We calculate the SMC by first finding the following:

- M_{00}: The number of words flagged as poor quality with any tag by both researchers
- M_{01}: The number of words that were flagged as poor quality by researcher 0 and **not** flagged by researcher 1
- M_{10}: The number of words that were flagged as poor quality by researcher 1 and **not** flagged by researcher 0
- M_{11}: The number of words that were not flagged by either researcher

We can then calculate the SMC using:

$$SMC = \frac{M_{00} + M_{11}}{M_{00} + M_{01} + M_{10} + M_{11}} \tag{4.1}$$

We can calculate this for each stimulus, for each system, and for the entire experiment and use this to identify agreement and explore where we disagreed and discuss potential causes. First, we looked at individual words. We looked at the average SMC for each word by system.

When rating our FastText, we had an average SMC of 72.66% and on our system we had an average SMC of 70.69%. This is lower than our overall agreements; when looking at all agreement and disagreement for FastText we have an SMC of 81.29%

Table 4.16: Data comparing how researchers tagged poor quality words for each tag in **our system** .

	O	R	S	D	U	Oth
Total	7	43	8	105	113	11
Researcher 1	3	20	5	61	37	3
Researcher 2	4	23	3	44	76	8
Agreed	5	31	5	81	82	8
Disagreed	2	12	3	24	31	3
% of Disagreed	2.67%	16.00%	4.00%	32.00%	41.33%	4.00%
% of Total	2.44%	14.98%	2.79%	36.59%	39.37%	3.83%

and our system has an agreement of 78.57% with a combined agreement of 80.14%. Overall this implies that we disagreed on the rating of about 20% of terms in our dataset with similar agreement across both systems and for any given word there is approximately 30% disagreement on which words are poor quality. This shows us that while the results are far more consistent than random, there is room for additional exploration into the reasons behind the 20-30% disagreement.

4.3.3 Problems with the Quality Indicators

To identify potential problems with our tags that could lead to our 20-30% disagreement we looked at how each of us used each tag and where tags were disproportionately used when we disagreed. While some amount of disagreement can be explained by user error when clicking hundreds tags, the breakdown of tags shows us additional problems with specific tags that offer us room for future work. Tables 4.16 and 4.17 show the data for disagreement across each individual system.

Table 4.16 shows mostly similar usage in tags across the dataset for our system. There is some discrepancy in repetition, synonymous words, and unrelated words, but not to the extent of those in Table 4.17 which shows the FastText disagreement. We speculate that the increased disagreement in FastText comes from the much higher quantity of issues which better highlights possible problems in our quality indicators. We will look at some examples of disagreement for each tag and discuss potential

Table 4.17: Data comparing how researchers tagged poor quality words for each tag in **FastText**.

	O	R	S	D	U	Oth
Total	104	239	121	140	30	78
Researcher 1	50	155	39	103	11	23
Researcher 2	54	84	82	37	19	55
Agreed	100	225	117	112	22	71
Disagreed	4	14	4	28	8	7
% of Disagreed	6.15%	21.54%	6.15%	43.08%	12.31%	10.77%
% of Total	14.61%	33.57%	16.99%	19.66%	4.21%	10.96%

issues with the indicators.

The synonymous tag was an issue in our system more than that of FastText. Reviewing some of the words in our system that were disagreed upon shows some of the different issues with the synonymous tag. One example of disagreement is when the stimulus *flower* was recommended the word *bloom*. One rater flagged it as synonymous, likely interpreting the part of speech as a *verb* while the other likely interpreted the word as a *noun*. This part-of-speech interpretation highlights the subjectivity of each rating and how much influence the rater's vocabulary and priming may affect how they rate words. We also see the stimulus *slaughter* having multiple disagreed tags, with *massacre* flagged by one rater as *synonymous* and as *obvious* by the other. The ambiguity between obvious and synonymous tags is difficult to account for; however, this shows the importance of looking at the poor quality words in aggregate as opposed to tag-by-tag as the word was still tagged by both researchers. Where the two raters fully disagreed was with the recommendation *rout*, which one flagged as synonymous to *slaughter*. This could again be an example of interpretation issues. We also see a disagreement with the stimulus *secretary*, where one rater marked it as *synonymous* and the other rated it as *repetitive*. This is primarily an issue with the *repetitive* tag.

The repetitive tag is likely an issue of policy when rating more-so than the tag itself. For example, one researcher would only flag words as repetitive if there were more than two instances, whereas one might have flagged any two similar words as being repetitive to one-another; for example, we see *soldier* and *warrior* flagged as

repetitive in the same list by one researcher and not by another. While this may just be a matter of interpretation it could also be seen as a policy issue as only those two were flagged as repetitive. We also see this as a matter of interpreting the paradigm of the word, such as in the case of words recommended for the word *secretary*. Here we see one researcher flagging *duke* and *marquis* as repetitive, while the other flagged both of those in addition to *minister* and *clerk*. It is possible that one researcher saw the four words as being similar to one-another by their occupation while the other excluded minister and clerk as those occupations are not similar enough to the others. The granularity of similarity and forming a consistent policy are the primary issues with repetition.

By far the biggest disagreements were with the *dull* and *unrelated* tags which share similar problems of policy and interpretation to one-another. Being tagged as simply dull or unrelated does not entirely imply that it is a poor quality recommendation; for example, one researcher tagged *golden* as unrelated to the stimulus *wing*. While golden has no inherent connection to wing, it is *compatible* with it and can evoke some positive imagery. This issue of unrelated, yet compatible shows up in multiple example, such as *slaughter* and *visitor*, *lust* and *snoop*, *rich* and *neighbour*, or *card* and *amuse*. Each of these may seem immediately unrelated and could be tagged but their compatibility with the word to potentially prompt new imagery means that they can be suitable recommendations. We see this in dull words as well, where words recommended in isolation may be dull but can offer some interesting imagery when juxtaposed with either the stimulus or other words in the recommendations list. For example, we see *self* and *sense* flagged as dull when recommended for the stimulus *weakness*. While self and sense may not evoke any topics or imagery on their own, when juxtaposed with the stimulus there is potential for new influences. While each of these tags has some problems, we can identify a few potential changes to test in future work.

4.3.4 Updating the Quality Indicators

When we review the current disagreements for word quality indicators we see some recurring issues:

- Policy: Each tag requires a strict implementation strategy for the raters to be consistent

- Vocabulary: The contexts that the rater has encountered these terms, if at all, will influence how they relate the word to other recommendations
- Interpretation: In addition to vocabulary, features such as part-of-speech and tense are mostly removed from the words and some raters may account for different interpretations of the recommendation

For extended user tests it is important to account for each of these possible issues in rating. For repetitive words, it is important to select a policy and stick with it; for example, if two recommendations are obviously related to one another, flag them as repetitive and do this for each possible pair of repetitive words. It may also be valuable to have policies regarding consistent flagging of multiple tags by asking the researchers to tag a word with every possible tag they can think of or to consistently only tag it with the single most problematic issue, depending on whether the purpose of the user study is to identify poor quality words or to test the efficacy of the quality indicators.

To account for vocabulary it may be required to perform some kind of vocabulary test to ensure that users are at a minimum level of English fluency; however, given that our system intentionally looks for somewhat obscure terms it may also be an issue of application. For example, a user which is using our system with a lower level of English fluency may desire less obscure word recommendations and being able to target words that are obscure may help to adjust the system as needed. We may also require a tag for when people are uncertain of a word's meaning so that it is not considered either *good* or *bad* quality when the rater is simply uncertain. While being able to discard a result as uncertain provides us with fewer false positives, we may still encounter issues of vocabulary in interpretation even if the user knows the meaning of the word.

Addressing interpretation is the most difficult problem as our system removes context from each word leaving the rater to be primed by their past experiences. While a rater may know the meanings of two words well, such as *slaughter* and *soldier*, it is likely that past experiences may influence their interpretation of how the words are connected. This would certainly be the case in controversial topics and when a rater has deep, domain specific knowledge that other raters may not have. While it is unclear how to completely deal with vocabulary and interpretation at the individual scale, it is possible that these will not be issues when a large number of participants perform the ratings. In addition to the connection between terms from

past experience, it is also possible that some raters will simply interpret the tense or part-of-speech differently from other raters. This could potentially be solved by showing multiple forms of the stimulus, such as showing *flower, flowering, flowered,* and *flowers* rather than simply *flower*. This offers the rater a much wider range of possible interpretations to consider and could reduce the disagreement for misinterpreted associations based purely on a lack of context.

While these fixes are purely speculation, these word quality indicators can be a valuable starting point for future creativity research and these specific issues are worth future exploration. For future testing of these tags, it is important to have strong policies over when a word should be tagged and with how many tags each word should have. To better analyze the effect of vocabulary on ratings it would be valuable to test the rater's vocabulary before they begin rating. Finally, offering some additional context for stimuli may help the raters think about the recommendations in different ways that will make their thinking more consistent to other raters who may be approaching it with a different initial context.

4.4 Conclusion

Each component of the system interacts with the other components in meaningful ways that require lots of testing and configuration to properly evaluate. With the configurations we tested, we were able to show that the selection of a domain appropriate corpus leads to domain appropriate associations that may be more valuable as creative influences than the standard semantic associations built by large, non-fiction corpora such as Google News. In addition to corpus selection, we were able to compare the qualities of sentence segmentation and TextTile segmentation, showing that while both are able to generate evocative and related results, TextTile is able to generate very evocative influences at the cost of also recommending comparatively many unrelated words. We showed that different collocation ranking schemes can generate very different results and that further exploration is needed to identify the qualities that define the different schemes.

We also looked into various filters, from basic blacklists to novel feature based filters that use emotional scores to successfully remove dull words and replace them with more evocative recommendations. We also made use of WordNet to successfully remove suggestions with obvious relationships to the stimulus to improve the overall

quality of the list and reduce repetitiveness. Making use of the different elements of our pipeline, we were able to begin exploration of recommendations generated by multiple stimuli and found that our methods produce diverse and interesting results with multiple stimuli, but also show that incorporating more than two stimuli or allowing words to be connected by very remote associations will reduce the quality of the output.

Finally, we developed a set of indicators based on our creative goals that give us a heuristic to evaluate recommended wordlists and word recommendation systems for creative tasks. Using these heuristics we showed that our system consistently outperforms both random association and FastText and that our system seems to produce lower quality recommendations when the stimulus is itself a poor quality word. We showed that we were able to achieve an 80% overall consistency between raters and identified potential points of issue that could be addressed and tested in future work. User studies are required to validate the indicators as an accurate measure of a wordlist's ability to enhance creativity.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this work, we explored word associations as a tool to aid creator's with the blank page problem for spatial storytelling when worldbuilding in games and novels. By using principles from different creative fields we began to understand the qualities that make certain types of word associations empower a creator's creativity. We hypothesized that traditional document analysis and word association work would fail to inspire creativity compared to associations that were selected to be remote and evocative. Using these principles we built a set of word quality indicators that were used to show that our system can produce more evocative associations than alternative methods by making use of a literary corpus, novel word filters based on emotional sentiment, and paragraph-scale segmentation using TextTile to build remote word associations.

Our primary contributions include:

- Defining a set of word quality indicators that help quantify a word association algorithm's creative potential
- Using novel features that remove words based on emotional sentiment and predefined relationships with the stimuli to recommend more evocative and remote words
- Improved the quality of creative word recommendations by using a domain-specific corpus

- Made progress toward showing how multiple stimuli can be used to generate relevant, yet novel themes
- Creating a creative theme recommendation system that consistently performs better than systems built for traditional word associations

By using a system pipeline inspired by the creative process we were able to produce a system which provides external, domain specific influences to a creator and help better understand what goes into creating beneficial, creative word associations for creative tasks.

We showed that making use of a literary corpus will provide more literary associations and that segmenting documents by multiple paragraphs can find remote and interesting thematic associations that can not occur through smaller segments at the cost of promoting some irrelevant terms. By combining our large segments with simple weighting and distance calculations we were able to produce remote yet relevant associations that are ideal for our creative goals.

To evaluate the quality of our wordlists according to our creative goals, we developed a set of word quality indicators that provided us with a simple heuristic to determine when wordlist and word recommendation system was of poor quality for creative purposes. When testing methods with these heuristics, we found that our work consistently outperformed the FastText algorithm, an algorithm that finds subtextual relations for text classification and other applications.

5.2 Future Work

Our system is capable of providing creative recommendations when compared to alternative systems; however, there is additional room for improving recommendations, extra validation, and various systems that we made early progress on testing but ultimately were unable to extensively explore. This section reviews some of the steps we have made toward continuing this research in different different directions and continued improvement.

5.2.1 User Studies

Our word quality indicators are based on creative principles and our own experiences reviewing hundreds of wordlists from different systems. While we believe these indicators do provide an excellent proxy metric for the overall quality of wordlists and word association systems, we also recognize that the results of continued testing would be bolstered by running user studies to validate the quality of the different indicators.

The goal of these user studies would be:

1. Evaluate the participant consistency when flagging words in different lists using our indicators
2. Validate that our indicators can be used to evaluate when word associations will help with the blank page problem
3. Evaluate which indicators may be more likely to indicate flaws in a system rather than a wordlist, and to what extent
4. Test the efficacy multi-stimuli recommendations for populating words when doing long term worldbuilding

We have started the design of the first user studie to begin testing how valid our word quality indicators are for practical applications of our system. As a pilot study, described in Chapter 3, we have built a survey which shows participants wordlists from FastText, random wordlists, and our system using cosine distances.

Additional studies can target individual features; for example, we could provide users with our internal wordlist indicator application, described in Chapter 3, to test the consistency of our indicators across many diverse participants. We can then use the results from that study to power a second study which focuses exclusively on testing how certain wordlists validated to have distinct features will help in creative writing tasks. These studies would be beneficial outside of our own project, helping researchers better understand what qualities of word associations best bolster creativity in other domains, such as open-ended problem solving.

5.2.2 System Component Analysis

This project covered a broad set of systems and evaluated each against possible alternatives. This broad evaluation prevented very in depth analysis of any single

component or the interactions between individual components. If our indicators are a valid proxy metric for our creative goals, we can use them to evaluate different system configurations and different system outputs. One of the first components we would like to explore further is the corpus selection.

The corpus selection represents the domain that the system can use to generate influences. Very early into the project we were able to use author's bibliographies to generate recommendations that were specific to the author. These author specific associated showed us that the corpus did have a lot of influence over the resulting recommendations; for example, *blood* might recommend dark themes when the system is trained on *Edgar Allen Poe* but would generate more familial themes when trained on *Jane Austen*. While these recommendations were very early and simplistic, it helped to validate that we needed to explore alternatives to Wikipedia, CommonCrawl, and Google News. This suggest we could find larger corpora than author bibliography and potentially have more specific domains than our *literary* tests, such as era or genre. Creativity research suggests that computers can act as multi-domain influencers that would otherwise not be thought of by the creator [29] and making the recommendations more specific to certain domains could help emphasize the different domains the system is taking inspiration from.

In addition to testing the corpus, we would like to explore segmentation techniques further. We know that the ranking schemes are sensitive to the segmentation technique, and we would need to compare results for additional segmentation methods to each of the ranking schemes. We have seen that sentence segmentation can produce evocative themes at the cost of losing some high quality remote associations and that TextTiling provided high quality and novel recommendations; however, it is unclear if one method performs consistently better for practical purposes than the other. There are many possible segmentation techniques to test, such as bigram segmentation, trigram segmentation, or single paragraph segmentation. A better understanding of how our indicators can accurately measure a list's creative potential would be required to properly analyze these different techniques.

5.2.3 Recommendations from Multiple Stimuli and Visualizing Themes

While it is important to better understand single word relations before further investigating scenarios with multiple stimuli, early results using the mean vector of the

stimuli subset were promising. Toivonen et al. discuss using methods such as harmonic mean and term frequency penalization to find optimal relations to the stimuli set [55]. We also produced good results when working with TF-IDF and random 2-stimuli subsets. Our early prototypes suggested that visual representation of the relationships may play an important role in how the list can bolster creativity.

During informal tests with an early mapping prototype, we were able to visualize how different sets of words could be used to generalize the themes of a geographical region. When spatial concerns are taken into account, the importance of distinct themes becomes much more evident. This may imply that recommendations which are generated to represent geographical themes must be more closely associated than those for blank-page storytelling. Not only could more work be done to investigate how users perceive words to associate with multiple stimuli, but it is worth exploring how the visual representation and user intentions modify a user's perception.

One early visualization we worked with was drawing connected themes across a map. When two themes were closely connected in different locations, a line could be drawn between those locations to indicate to the creator that their may be a viable story connection between them and could prompt new stories either in those locations or between them. We also worked with visualizing the connection of themes using *word paths*. These paths would take two unrelated themes and make a "path" from one theme to the next. An example path from our old system going between *love* and *war* is: *love, life, people, government, war*. These interim themes may help users visualize stories in ways they had not thought of, and these paths could themselves be used as stimuli for multi-stimuli recommendations.

List of References

[1] 8 Dudes in a Garage AB, "The Elder Scrolls V: Skyrim | IGDB.com," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: https://www.igdb.com/games/the-elder-scrolls-v-skyrim/credits

[2] S. Acar and M. A. Runco, "Assessing associative distance among ideas elicited by tests of divergent thinking," *Creativity Research Journal*, vol. 26, no. 2, pp. 229–238, 2014.

[3] T. Adams and Z. Adams, "Dwarf Fortress (PC Game)," 2006.

[4] J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2003, pp. 314–321.

[5] C. R. Aragon and A. Williams, "Collaborative creativity: A complex systems model with distributed affect," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1875–1884. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979214

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[7] J. G. Brown, "Creativity and problem-solving," *Marq. L. Rev.*, vol. 87, p. 697, 2003.

[8] Y.-T. Chen and M. C. Chen, "Using chi-square statistics to measure similarities for text categorization," *Expert systems with applications*, vol. 38, no. 4, pp. 3085–3090, 2011.

[9] S. Colton, "The Painting Fool: Stories from building an automated painter," in *Computers and creativity*. Springer, 2012, pp. 3–38.

[10] E. S. W. Contributors, "Categories," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: http://elderscrolls.wikia.com/wiki/Special:Categories

[11] M. Cook, S. Colton, and J. Gow, "Automating game design in three dimensions," in *Proceedings of the AISB Symposium on AI and Games*, 2014, pp. 20–24.

[12] M. Cook, S. Colton, and A. Pease, "Aesthetic considerations for automated platformer design." in *AIIDE*, 2012.

[13] C. Crawford, *Chris Crawford on interactive storytelling*, 1st ed. Berkeley, CA: New Riders, 2005.

[14] J. Doran and I. Parberry, "A prototype quest generator based on a structural analysis of quests from four MMORPGs," in *Proceedings of the 2Nd International Workshop on Procedural Content Generation in Games*, ser. PCGames '11. New York, NY, USA: ACM, 2011, pp. 1:1–1:8. [Online]. Available: http://doi.acm.org/10.1145/2000919.2000920

[15] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Computational linguistics*, vol. 19, no. 1, pp. 61–74, 1993.

[16] J. M. Epstein, MIT CogNet, and 2050 Project, *Growing artificial societies: social science from the bottom up.* Cambridge, Massachusetts: The MIT Press, 1996.

[17] Explosion AI, "Annotation Specifications · spaCy API Documentation," Nov 2018, [Online; accessed 2. Dec. 2018]. [Online]. Available: https://spacy.io/api/annotation

[18] Freehold Games, "Caves of Qud (PC Game)," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: http://www.freeholdgames.com/press/sheet.php?p=caves_of_qud

[19] I. L. Götz, "On defining creativity," *The Journal of Aesthetics and Art Criticism*, vol. 39, no. 3, pp. 297–301, 1981.

[20] O. Gross, H. Toivonen, J. M. Toivanen, and A. Valitutti, "Lexical creativity from word associations," in *Knowledge, Information and Creativity Support Systems (KICSS), 2012 Seventh International Conference on.* IEEE, 2012, pp. 35–42.

[21] S. Gupta, A. Nenkova, and D. Jurafsky, "Measuring importance and query relevance in topic-focused multi-document summarization," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions.* Association for Computational Linguistics, 2007, pp. 193–196.

[22] L. Hall-Stigerts, "Open-world games' history of creativity and controversy | Big Fish Blog," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: https://www.bigfishgames.com/blog/history-of-open-world-games

[23] M. A. Hearst, "TextTiling: Segmenting text into multi-paragraph subtopic passages," *Computational linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[24] Hello Games, "No Man's Sky (PC Game)," 2016.

[25] R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: a formal approach to game design and game research," in *Proceedings of the AAAI Workshop on Challenges in Game AI*, vol. 4, no. 1, 2004, p. 1722.

[26] N. B. Huntemann and M. T. Payne, *Joystick soldiers: The politics of play in military video games.* Routledge, 2009.

[27] H. Jenkins, "Game design as narrative," *Computer*, vol. 44, p. 53, 2004.

[28] M. Johnson, "Ultima Ratio Regum (PC Game)," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: http://www.ultimaratioregum.co.uk/game/info

[29] J. Koch, "Design implications for designing with a collaborative AI," in *AAAI Spring Symposium Series, Designing the User Experience of Machine Learning Systems*, 2017.

[30] R. Koster, *Theory of fun for game design*. O'Reilly Media, Inc., 2013.

[31] T. Krzywinska, "Blood scythes, festivals, quests, and backstories: World creation and rhetorics of myth in World of Warcraft," *Games and Culture*, vol. 1, no. 4, pp. 383–396, 2006.

[32] S. Lahiri, "Complexity of Word Collocation Networks: A Preliminary Structural Analysis," in *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, April 2014, pp. 96–105. [Online]. Available: http://www.aclweb.org/anthology/E14-3011

[33] W. LLC, "Worldspinner: Fantasy map making and world building," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: https://worldspinner.com

[34] C. D. Manning, P. Raghavan, and H. Schütze, *Scoring, term weighting, and the vector space model*. Cambridge University Press, 2008, p. 100123.

[35] S. Mednick, "The associative basis of the creative process." *Psychological review*, vol. 69, no. 3, p. 220, 1962.

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[37] D. Mitra, "Computational creativity: Three generations of research and beyond." in *AAAI Spring Symposium: Creative Intelligent Systems*, 2008, pp. 47–52.

[38] R. C. Moore, "On log-likelihood-ratios and the significance of rare events," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[39] R. Moss, "Roam free: A history of open-world gaming," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: https://arstechnica.com/gaming/2017/03/youre-now-free-to-move-about-vice-city-a-history-of-open-world-gaming

[40] R. C. Muller, "Enhancing creativity, innovation and cooperation," *AI & Society*, vol. 7, no. 1, pp. 4–39, 1993.

[41] M. O'Leary, "Ketchum: Build collections of words," Jan 2018, [Online; accessed 9. Dec. 2018]. [Online]. Available: https://github.com/mewo2/ketchum

[42] R. S. Pfeiffer, "The scientific concept of creativity," *Educational Theory*, vol. 29, no. 2, pp. 129–137, 1979.

[43] T. Pickering and A. Jordanous, "Applying narrative theory to aid unexpectedness in a self-evaluative story generation system," in *8th International Conference on Computational Creativity*, May 2017, pp. 213–220. [Online]. Available: https://kar.kent.ac.uk/61661/

[44] M. A. Picucci, "When video games tell stories:a model of video game narrative architectures," *Caracteres: Estudios Culturales y Crticos de la Esfera Digital*, vol. 3, no. 2, pp. 99–116, 2014.

[45] R. Plutchik, "The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice," *American scientist*, vol. 89, no. 4, pp. 344–350, 2001.

[46] T. O. Ring.net, "Interviews — Gary Gygax - The Creator of Dungeons and Dragons," May 2000, [Online; accessed 30. Jan. 2019]. [Online]. Available: http://archives.theonering.net/features/interviews/gary_gygax.html

[47] Y. A. Rotmistrov, "About the Project - Word Associations Network," Nov 2018, [Online; accessed 30. Nov. 2018]. [Online]. Available: https://wordassociations.net/en/about

[48] J. Ryan, A. J. Summerville, M. Mateas, and N. Wardrip-Fruin, "Translating player dialogue into meaning representations using LSTMs," in *International Conference on Intelligent Virtual Agents*. Springer, 2016, pp. 383–386.

[49] K. R. Scherer and H. G. Wallbott, "Evidence for universality and cultural variation of differential emotion response patterning," *Journal of Personality and Social Psychology*, vol. 66, no. 2, pp. 310–328, 1994.

[50] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[51] J. Staiano and M. Guerini, "DepecheMood: a lexicon for emotion analysis from crowd-annotated news," *arXiv preprint arXiv:1405.1605*, 2014.

[52] V. C. Storey, "Understanding semantic relationships," *The VLDB Journal*, vol. 2, no. 4, pp. 455–488, 1993.

[53] A. Summerville and M. Mateas, "Super Mario as a string: Platformer level generation via LSTMs," *arXiv preprint arXiv:1603.00930*, 2016.

[54] W. Tatarkiewicz, *A History of Six Ideas: An Essay in Aesthetics*, ser. Melbourne International Philosophy Series. Springer Netherlands, 2012. [Online]. Available: https://books.google.ca/books?id=zPrxCAAAQBAJ

[55] H. Toivonen, O. Gross, J. M. Toivanen, and A. Valitutti, "On creative uses of word associations," in *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*. Springer, 2013, pp. 17–24.

[56] E. P. Torrance, "Scientific views of creativity and factors affecting its growth," *Daedalus*, vol. 94, no. 3, pp. 663–681, 1965.

[57] P. University, "WordNet | A Lexical Database for English," Dec 2018, [Online; accessed 3. Dec. 2018]. [Online]. Available: https://wordnet.princeton.edu

[58] A. B. Warriner, V. Kuperman, and M. Brysbaert, "Norms of valence, arousal, and dominance for 13,915 English lemmas," *Behavior Research Methods*, vol. 45, no. 4, pp. 1191–1207, Dec 2013. [Online]. Available: https://doi.org/10.3758/s13428-012-0314-x

[59] M. J. Wolf, *Building imaginary worlds: The theory and history of subcreation*. Routledge, 2014.

[60] I. Worlds, "Inkarnate," Nov 2018, [Online; accessed 29. Nov. 2018]. [Online]. Available: https://inkarnate.com

[61] D. Yu and A. Hull, "Spelunky (PC Game)," 2009.

Appendix A

Differences Between Internal Ranking Algorithms

Our system was tested using three different algorithms each with thirty-two sets of parameters across a total of thirty-three different words. In the table below we show all words that were unique to the combined set of all of our parameter testing for each algorithm. The algorithms and their parameters have been described in Chapter 3. For brevity only a select number of words appears in this table. Note that that each wordlist is the set of all unique words that appeared for every parameter tested and as such these lists should not be used as a test for the quality of associations.

Table A.1: Words unique to each ranking method for the stimulus: **flower**

Ranking Method	Unique Words Related to **flower**
Cosine Distance	bright, delicious, leaf, shade, shady, sunny, sunshine, weed
LLR	fruit, joy, song, wonderful
TF-IDF	grass, kiss, poem, valley

Table A.2: Words unique to each ranking method for the stimulus: **echo**

Ranking Method	Unique Words Related to **echo**
Cosine Distance	babble, burst, ear, hush, sing, star, strain, sullen, tread
LLR	distant, shade, sigh, tale
TF-IDF	gun, poem, roar, valley

Table A.3: Words unique to each ranking method for the stimulus: **grave**

Ranking Method	Unique Words Related to **grave**
Cosine Distance	careless, cast, gentle, marvel, proud, shrink, sober, solemn, tongue, touch
LLR	ago, beneath, drink, evil, teach
TF-IDF	bless, gate, gaze, poet, soldier

Table A.4: Words unique to each ranking method for the stimulus: **lust**

Ranking Method	Unique Words Related to **lust**
Cosine Distance	beget, bondage, carnal, glorify, glory, heavenly, sinful, snare, unclean
LLR	base, crime
TF-IDF	angel, flee

Table A.5: Words unique to each ranking method for the stimulus: **wander**

Ranking Method	Unique Words Related to **wander**
Cosine Distance	dreary, dull, fill, haunt, lonely, restless, sight, solitary, stray, wand
LLR	desert, dream, drink, grave, leaf, pleasant, wave
TF-IDF	gate, grass, music, poet, savage, song, valley

Table A.6: Words unique to each ranking method for the stimulus: **ardour**

Ranking Method	Unique Words Related to **ardour**
Cosine Distance	ardent, cherish, destiny, devote, inspire, manly, pride, spirit, triumph
LLR	admire, cool, genius, glory, hero
TF-IDF	affection, courage, happiness, lover, poem

Table A.7: Words unique to each ranking method for the stimulus: **pale**

Ranking Method	Unique Words Related to **pale**
Cosine Distance	bend, brighten, forehead, gaze, hair, lip, star
LLR	behold, golden, shade, stone
TF-IDF	color, exclaim, kiss, moon, tremble

Table A.8: Words unique to each ranking method for the stimulus: **ghost**

Ranking Method	Unique Words Related to **ghost**
Cosine Distance	cast, dead, demon, dream, dumb, fearful, forth, grave, marvel, solemn
LLR	vision
TF-IDF	savage

Table A.9: Words unique to each ranking method for the stimulus: **prisoner**

Ranking Method	Unique Words Related to **prisoner**
Cosine Distance	charge, command, guard, muster, officer, report, rescue, safety, soldier, threaten
LLR	iron, savage, taking
TF-IDF	duke, murder, wound

Table A.10: Words unique to each ranking method for the stimulus: **divine**

Ranking Method	Unique Words Related to **divine**
Cosine Distance	earthly, evil, greatness, infinite, inspire, purify, reverence, spirit, sublime
LLR	genius, imagination, praise, vision
TF-IDF	angel, poem, sorrow, verse

Table A.11: Words unique to each ranking method for the stimulus: **prison**

Ranking Method	Unique Words Related to **prison**
Cosine Distance	death, fear, fit, hop, jail, rob, serve, suffer, threaten
LLR	charge, joy, wear, yard
TF-IDF	bless, duke, poem, slave

Table A.12: Words which were common between at least two of the three ranking methods used

Ranking Method	Common Words
flower	bloom, lovely, fairy, seed, bee, blossom
echo	noise, tremble, laughter, moon, verse, murmur
grave	bury, song, poem, sorrow, tale
slaughter	wound, flee, hell, slay, massacre, destruction, cattle, curse
lust	throne, temptation, tempt, hell, fame, cruelty, wick, gospel
wander	poem, journey, moon
parade	pomp, military, regiment, drill, uniform, ceremony, cavalry, sergeant, drum
spirit	behold, passion, virtue, flesh
tower	cathedral, distant, moon, spread, golden, lofty
ardour	emotion, military, poetry, verse, enthusiasm
pale	brow, grey, cheek, gleam, countenance, murmur
card	visitor, player, luck, invitation, tea, guest, trick, gamble
power	passion, virtue, imagination, destroy, poet, glory
ghost	terror, poem, glory, hell, murder, flesh, wolf, moon, haunt
wing	golden, beast, feather, poem, pigeon, flutter
prisoner	surrender, arrest, regiment, earl, troop, crime, trial
rich	color, labour, silver, valley, golden, enjoy
pocket	thrust, coat, pick, waistcoat, pistol
paper	poem, dinner, report, print, writer, colonel, thank
opinion	minister, notion, authority, virtue, poem, favour, writer
divine	mankind, sacrifice, poetry, mortal, spiritual, hell
fortune	favour, poem, marriage, virtue
green	color, lovely, valley, grass, shade, sand, shin, golden, meadow
weakness	courage, passion, virtue, folly, pride, sorrow, glory
secretary	minister, majesty, military, earl, council, poem, clerk, duke
prison	murder, confine, arrest, crime, trial, hell
improve	acquaintance, habit, poet, breed, selection
sister	affection, bless, liked, poem, colonel
majority	minister, example, council, title, duke, slave, mankind, favour
creation	origin, organic, angel, universe, mankind, creator, literature, selection
perceive	quality, presently, behold, command, imagine

Appendix B

List of Poor Part-of-Speech Tags

Below is a list of part of speed tags that were determined to be frequently associated with poor quality words for the purposes of blacklisting. These tags are used in the Python natural language processing library SpaCy. The tags and descriptions are from the official documentation and have been trimmed to only include the tags that are in the blacklist [17].

Tag	POS	Description
-LRB-	PUNCT	left round bracket
-RRB-	PUNCT	right round bracket
,	PUNCT	punctuation mark, comma
:	PUNCT	punctuation mark, colon or ellipsis
.	PUNCT	punctuation mark, sentence closer
’	PUNCT	closing quotation mark
””	PUNCT	closing quotation mark
#	SYM	symbol, number sign
“	PUNCT	opening quotation mark
$	SYM	symbol, currency
ADD	X	email
CC	CONJ	conjunction, coordinating

CD	NUM	cardinal number
DT	DET	
EX	ADV	existential there
FW	X	foreign word
GW	X	additional word in multi-word expression
HVS	VERB	forms of "have"
HYPH	PUNCT	punctuation mark, hyphen
IN	ADP	conjunction, subordinating or preposition
LS	PUNCT	list item marker
MD	VERB	verb, modal auxiliary
NFP	PUNCT	superfluous punctuation
NNP	PROPN	noun, proper singular
NNPS	PROPN	noun, proper plural
PDT	ADJ	predeterminer
POS	PART	possessive ending
PRP	PRON	pronoun, personal
PRP$	ADJ	pronoun, possessive
RP	PART	adverb, particle
_SP	SPACE	space
SYM	SYM	symbol
TO	PART	infinitival to
UH	INTJ	interjection
WDT	ADJ	wh-determiner
WP	NOUN	wh-pronoun, personal

WP$	ADJ	wh-pronoun, possessive
WRB	ADV	wh-adverb
XX	X	unknown

Appendix C

Comparisons to Other Systems

Table C.1: Comparison of results generated by our system, using cosine distance and a 0.175 valence threshold to FastText and Word Associations Network. The results from WAN were chosen by looking at the first word in each part-of-speech until ten words were chosen.

System	Stimulus	Recommendations
Ours	echo	ear, strain, hush, star, sing, sullen, burst, murmur, babble, tread
FastText	echo	detune, stero, rasping, transfix, lightpen, pushbroom, rasp, spirant, digust, confabulate
WordAssociations	echo	footstep, shrill, boom, faintly, shriek, vaulted, ring, sentiment, eerie, thunder
Ours	flower	leaf, shade, sunny, sunshine, bright, bloom, weed, delicious, shady, lovely
FastText	flower	crocus, cornflower, peony, tulip, bouquet, garland, dewdrop, vase, leafage, bottlebrush
WordAssociations	flower	inflorescence, flowering, pluck, freely, petal, hardy, thrive, singly, centimeter, branched
Ours	grave	solemn, careless, cast, gentle, touch, sober, shrink, marvel, proud, tongue
FastText	grave	sepulcher, sorrowful, catacomb, stone, cromlech, posthole, disquiet, crypt, blood-shed, drear
WordAssociations	grave	epitaph, unmarked, inscribe, lowly, burial, marble, dig, digger, stone, bury
Ours	lust	heavenly, unclean, hell, glorify, beget, sinful, bondage, glory, snare, carnal
FastText	lust	greed, avarice, jealousy, debauchery, fornication, gluttony, madness, ugliness, cruelty, extravagance
WordAssociations	lust	adultery, wicked, inflame, lucian, sexual, gratify, flesh, evil, incite, revenge
Ours	slaughter	rout, slay, flee, wound, assault, destruction, rebel, soldier, warrior, massacre
FastText	slaughter	slaugher, carnage, massacre, killings, murder, dismemberment, decapitation, bloodletting, death, incineration
WordAssociations	slaughter	innocent, sacrificial, calve, cardcasse, humane, slay, eunuch, wholesale, route, butcher
Ours	wander	lonely, stray, haunt, restless, solitary, wand, dreary, fill, dull, sight
FastText	wander	drift, ruminate, leave, waylay, gather, espy, roust, hang, away, reck
WordAssociations	wander	gaze, fro, daze, idly, vagabond, hither, roam, blindly, maze, thither

Appendix D

FastText Data Using Our Whitelist

The following data was collected by running the fast text corpus through our initial filter. Each word had recommendations generated until ten could be provided and the number of removed words to achieve ten recommendations was recorded.

Table D.1: Sample of words generated by FastText that were run through the system and how many words needed to be filtered to obtain a top ten.

Stimulus	n Filtered	Sample Filtered Words (at most 5)
echo	16	detune, stero, pushbroom, spirant, digust
grave	12	posthole, whinstone, drear, unresting, perpetuall
parade	19	camp-out, wedding-themed, tickertape, face-painting, knees-up
spirit	13	self-will, good-neighborliness, open-mindedness, spirit-, heartedness
tower	5	gantry, headframe, firehouse, circuitboard, three-floor
card	26	turn-up, punch-out, chargeback, tear-out, sign-out
power	25	inluence, efficency, selfconfidence, poistion, connectiveness
ghost	10	haunter, hand-puppet, rat-catcher, shoe-shine, boogyman
wing	21	frontward, out-flank, tailstock, wing-play, flying-off
prisoner	11	supersoldier, child-minder, hostage-taker, foot-soldier, hospitaller
rich	49	not-so-rich, -rich, history-rich, income-poor, mega-wealthy
pocket	9	cubby-hole, shoebox, foldaway, daypack, slingback
paper	33	paper-and-ink, newsprint, half-sheet, waste-paper, newssheet
opinion	13	oppposition, clarifcation, controversion, discusstion, propostion
divine	8	God-bearing, spritual, supernal, God-made, life-giving
fortune	4	cival, life–and, benifit, enterprize
green	21	non-green, light-green, green-red, red-yellow, deep-green
weakness	7	over-confidence, mercilessness, needlessness, unreadiness, impotency
prison	17	watchhouse, stalag, almshouse, watch-house, rest-room
sister	18	sister-, sister-wife, step-brother, step-sister, half-sister
majority	70	majority-, non-third, out-number, people–even, poeople
creation	8	productization, restructuration, building-up, co-creation, implementation
perceive	12	misperceiving, instrumentalize, depersonalize, consious, misperceive
score	108	pointscorer, career-low, minus-2, team-best, 90-point
job	57	dayjob, pre-job, single-task, job–and, jobholder

Appendix E

Data from Heuristics Evaluation

Table E.1: Average number of words flagged as poor quality for all stimuli tested.

Stim	Ours		FastText		Random	
ardour	2	1	6	9	-	-
card	9	5	10	10	-	-
creation	1	2	7	10	-	-
divine	0	0	9	8	-	-
echo	2	3	7	9	8	10
flower	5	5	7	8	6	7
fortune	2	5	2	9	-	-
ghost	3	2	6	7	-	-
green	7	5	10	10	-	-
improve	2	2	10	10	-	-
lust	3	1	5	6	8	7
majority	6	7	9	10	-	-
opinion	4	0	7	10	-	-
pale	6	4	10	10	-	-
paper	8	10	9	9	-	-
parade	3	3	6	8	-	-
perceive	6	2	7	10	-	-
pocket	5	8	9	8	-	-
power	6	5	9	9	-	-
prison	4	2	8	9	-	-
prisoner	4	4	8	8	-	-
rich	2	3	10	10	-	-
secretary	5	5	10	9	-	-
sister	3	4	10	10	-	-
slaughter	2	6	8	9	8	6
tower	3	3	6	7	-	-
wander	2	2	7	3	-	-
weakness	2	2	8	8	-	-
wing	8	5	10	10	-	-