How Will it End? OPERA As an Approach to Prediction.

by

Jennifer L. Madiedo

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author ……………………………………………………………………………………………

Department of Electrical Engineering and Computer Science

May 11, 2020

Certified by ……………………………………………………………………………………

Randall Davis

Professor of Computer Science and Electrical Engineering

Thesis Supervisor

Accepted by ……………………………………………………………………………………

Katrina LaCurts

Chair, Master of Engineering Thesis Committee

How Will it End? OPERA As an Approach to Prediction

by

Jennifer L. Madiedo

Submitted to the Department of Electrical Engineering and Computer Science
on May 11, 2020, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Developing a computational account of human intelligence requires understanding and modeling human story understanding faculties, including the human ability to make predictions. When presented with a novel situation, people can generalize their past experiences and apply them to propose a reasonable prediction about potential outcomes of that situation. The goal of this work is to develop a novel approach to develop this reasoning capability.

We present character alignment as a new approach to outcome prediction. We claim that if you can capture a character's behavior and motivations in a story, you can use that information to make predictions about a similar character in the future. Inspired by work on reasoning by analogy and story alignment, our system uses the character alignment approach to create representations of characters as it reads stories and uses those representations to predict what a character in a new story might do. Our representations are action oriented, i.e., focused on how a given character interacts with other characters and the specific actions they take. We claim that this focus on character representations is valuable in part for its efficiency: It enables reasoning with only key parts of stories rather than the entire story.

To demonstrate this approach in action, we designed and developed OPERA, a system that makes predictions about story outcomes. Built on top of the Genesis system, OPERA gathers information about characters in the short stories that it reads (approx. 60-70 sentences long). For each story, the system creates representations for each character by extracting information about them and their interactions. This information includes their desires, the actions they take, and the goals they achieve. When prompted with a specific character in the story, the system uses its representation of characters to predict the outcome for that character.

This method has advantages over existing approaches, such as story alignment, because it focuses on key parts of the story (specifically those in character representations) and can combine and generalize information from multiple sources. This work gives Genesis the ability to efficiently draw connections from past experiences, recognize similarities, and make informed predictions.

Thesis Supervisor: Randall Davis
Title: Professor of Computer Science and Electrical Engineering

Acknowledgements

Gracias Mami, for always listening when things did not work and offering reassuring words. For always assuring me that you knew I could do it, and that you knew I would figure it out. Gracias Papi, for being the motivator I did not always know I needed. For pushing me to be my best and strive for something I would be proud of. Gracias a los dos, for helping me achieve this dream. Thank you, Andrea, for reminding me to stop taking myself so seriously. For getting me to be silly and celebrate the little victories along the way.

Thank you, Dylan, for always meeting my panicked emails with calm assurance that everything would be okay. For knowing what to say and having the patience to help me figure out my problems when I did not know how to explain them. Thank you, Kimberle, for helping me become a better communicator. For the words of encouragement during stressful times. Thank you for helping me find the words and say the things.

Thank you, Professor Winston, for encouraging me to pursue my passions, for all great works start that way. For letting me ramble on for hours in your office as I found my way through my ideas, offering me coffee when I started to lose steam. For introducing me to AI and guiding me as I grew into my shoes as a researcher. Thank you for teaching me that I can do it, only I can do it, I cannot do it alone, and I never will if I take care of my people. Thank you for teaching me how to communicate and how to tell *my* story.

To Professor Davis, for taking me on without a second thought. For teaching me how to articulate my ideas, how to weather the roadblocks, and how to rise above. Thank you for the words of encouragement during the hard times, and the celebratory smiles during the good times. For helping me grow as a critic, a researcher, a teacher, and a person. Thank you for all the wisdom and the chocolate.

Contents

List of Figures

List of Tables

1 Vision

Work in artificial intelligence seeks both to understand what makes humans uniquely intelligent and to build systems that exhibit intelligent behavior. We believe that building computational models of human intelligence helps us progress in achieving both of these goals. We believe that what makes human intelligence unique among species is our ability to reason symbolically [1], which in turn enables us to understand stories—something no other species can do. As Winston articulates, "…we are symbolic story-understanders" [2], and as he maintains in his Strong Story Hypothesis, "The mechanisms that enable humans to tell, to understand, and to recombine stories separate our intelligence from that of other primates." [3]

This thesis builds on Winston's Genesis story understanding system, extending its computational model to include a key aspect of human intelligence: the ability to predict an outcome. Prediction can be accomplished by analyzing prior situations, drawing connections between the situations, matching similar situations, and generalizing to a current situation. There have been many previous approaches to building systems that can predict story outcomes; they fall into two broad categories: symbolic methods and statistical machine learning methods. Machine learning methods, such as deep learning and recurrent networks, focus on extracting specific features of the story – sentiment, context words, patterns etc. – to make predictions.

Despite advances and improvements in machine learning approaches, these techniques still suffer poor results in tasks that require higher-level reasoning and understanding. The 2017 Story Cloze Test [4], developed by a team of researchers across academia and industry as an empirical evaluation framework, asks a system to choose the correct ending for a four-sentence story. An evaluation of 11 state-of-the-art models showed that they achieved only about 50% accuracy. The team doing the evaluation noted that many of the current systems rely too heavily on statistical methods, to the point that the systems are simply learning how to "beat the test." We believe that systems need to have a deeper, richer semantic understanding of the stories in order to achieve better performance. We further believe that the deeper and richer understanding needed for tasks such as these is better suited to symbolic methods that model human reasoning.

One symbolic method that served as inspiration for the work described in this thesis is analogical reasoning. Reasoning analogically in support of prediction is based on answering 4 key questions:

1. How do you compare experiences?
2. How do you align similar experiences?
3. How to you evaluate those alignments?
4. How do you apply those alignments to the current situation?

We claim that by answering these questions in the context of story understanding, we can improve on statistical machine learning results when generating relevant predictions.

One approach to prediction is classed story alignment, an approach inspired by analogical reasoning. It calls for matching up entire stories to make predictions. We believe that approach is sub-optimal because it considers a lot of extraneous information that we claim is not needed for making predictions and that is computationally expensive. Fay [4] used story alignment between two stories at a time and showed that a brute force method with *n* plot units[1] had a worst-case runtime of $O(n!)$. He developed optimizations that cut the runtime to $O(n^2)$, but for only a single alignment. Fay's work is an example of how story alignment can be computationally expensive and is not easily scalable.

With these ideas in mind, we propose **character alignment:** a symbolic, analogical reasoning-

[1] A plot unit is syntactic unit for the story. Plot units are usually equivalent to individual sentences, except in the case of complex sentences, which are usually split up into multiple plot units.

based approach to prediction in story-based systems. **Character alignment** shifts the focus from the entire story to the individual characters themselves. We claim that if you can capture a character's behavior and motivations in a story, you can use that information to make predictions about a similar character in the future. In particular, the **character alignment** approach:

1. **Define** an action-oriented character representation made up of 3 core parts that capture a character's behaviors and motivations.
2. **Recognizes** potential character alignments, using a two-level matching system for pattern recognition and grouping amongst characters.
3. **Analyzes** the value of each potential alignment, using a weighting system that prioritizes the most relevant information and enables scoring of each alignment.
4. **Predicts** an outcome using a multi-level prediction pyramid that builds up the final prediction in stages.

To demonstrate and test the character alignment approach, we designed and developed OPERA (Outcome Prediction Enabled by chaRacter Alignment). OPERA is built as a module on top of the Genesis Story Understanding system[2]. OPERA reads stories to gather information, parsing them using Genesis and then creating representations for each character. When prompted with a specific character, OPERA triggers a set of experts to search for potential matches, then aligns the matches to each other, computing a score for each match. OPERA uses these matches to predict the outcome for the character.

The character alignment approach and the accompanying OPERA system were designed to model human intelligence and reasoning capabilities. There is no immense database of data; OPERA reads in one short story at a time and then prompts the user to either select a character for

[2] The Genesis Story Understanding System was developed in the Genesis Group at CSAIL by the late Prof. Patrick Winston and his graduate students. Genesis's capabilities continue to be expanded. See [2] for more information.

outcome prediction or select another story to read.

To test OPERA, we used a set of stories that the system reads to gain experience, and a small set of test stories with their endings removed. Some endings were removed earlier in the story versus later in the story, to determine how OPERA performs with more/less information about the target story. We show how OPERA's ability to combine information from multiple sources during character alignment helps it avoid source bias i.e., when your result is tied too closely to the sources you used as experiences

Consider the following example: OPERA has been given 3 stories to read for experience – *Hamlet, Macbeth, and Anastasia* – and is then given an abbreviated version of *The Lion King.* We terminated the story when Scar and Simba fight, which is the climax of the story, and ask OPERA to propose an outcome. Table 1 below shows potential results, including matches for the two main *Lion King* characters – Scar and Simba – against the characters from three other stories, as well as the potential outcome based on those matches.

Character	**Chosen Alignment and Score**	**Potential Outcome**
Scar	Scar ↔ Claudius: 36 Scar ↔ Macbeth: 34.3 Scar ↔ Rasputin: 28.5 Scar ↔ Hamlet: 17.6	Simba harms Scar, who becomes incapacitated
Simba	Simba ↔ Rasputin: 13 Simba ↔ Hamlet: 11 Simba ↔ Macbeth: 10 Simba ↔ Claudius: 9	Simba becomes king

Table 1: Example OPERA results given Hamlet, Anastasia, and Macbeth read as background, and abbreviated, terminated version of Lion King read last. Character alignment gathers information from multiple stories and combines it for a generalized outcome. The numbers in the **chosen alignment** column represent the confidence scores OPERA assigned to each of the alignments.

Character alignment focuses on combining information across all three stories. We can see in the table that both Scar and Simba are matched to multiple characters with varying levels of confidence. OPERA combines all those alignments and generalizes across them to predict the most likely outcome. Looking at Simba, this leads to the prediction that Simba will become king.[3]

While combining information across multiple stories can help reduce the noise of dissimilar character(s) that could lead to incorrect outcomes, could is also obscure the character(s) that would eventually lead to the desired outcome? We avoid this by careful attention to the scoring system for alignments, as well as the process for combining the scores when generalizing the alignments (See sections 4.3 and 4.4.) In short, the level of generalization that occurs depends highly on the scores of the alignments so that characters that are most similar will exert a much greater influence on the outcome than those that are not, even with the combination and generalization.

This example in table 1 above illustrates our method of character alignment, a novel approach for predicting outcomes in stories. This approach focuses on applying past experiences, drawing connections between experiences to provide a basis on which a prediction can be made. In the remainder of this document we motivate the work by explaining the role prediction plays in achieving human-level intelligence and review current approaches. We then lay out the conceptual framework underlying character alignment, based on work in analogical reasoning. We describe details of an implementation built as a module on top of the Genesis system. Finally, we present experimental results and discuss the contributions of our approach.

[3] See appendix for story summaries.

2 Prediction Comes Naturally to Humans. Machines Not So Much

Prediction is an innate ability that comes naturally to humans. As young children, prediction helps us learn faster and more efficiently, decreasing the cognitive load of all the new stimuli we receive [5]. As we get older, prediction plays into our daily decision making. Like many things that are cognitively simple for humans however, prediction is a difficult task for machines. This chapter explains why prediction is an important ability for intelligent machines. We examine the role of prediction in human cognition and show how the task naturally lends itself to story understanding. We describe the two common prediction tasks – selection and generation – and the approaches to solving them. Lastly, we review some existing machine learning and symbolic approaches.

2.1 Prediction Is Tied to Human Cognition

An ongoing branch of research in neuroscience is trying to understand the fundamental mechanism that leads to such rapid cognitive development in young children. This is especially evident in language acquisition, where children between the ages of 1 and 4 go from knowing a handful of words to a vocabulary of well over a thousand only 3 years later. Many studies have suggested that this expansive cognitive development may be due in part, to our "predictive brain." This idea posits prediction as one of the fundamental principles of neural computation, such that errors of prediction drive neural and cognitive processes as well as behavior. [6] [7]

Prediction enables us to reason about the world and expand our understanding based on the things we know. Let us look at language again. Studies have shown that children are constantly using prediction to learn new words and conjugations [5]. Apart from picking up words they hear around them, children are also forming their own internal model of the language and trying to learn the rules that lead to new words. This leads to a lot of generalization and testing of new (often incorrect) words or phrases. In some cases, mistakes are corrected, but many times, they are not. A common example of this is children overgeneralizing when learning plurals, as shown in figure 2.1:

```
dog->dogs
horse->horses
mouse->mouses
```

Figure 2.1: Example of common overgeneralization of plurality rule by children. Most children naturally learn that the plural of a word is obtained by adding -s or -es, leading them to believe that the plural of mouse is mouses [8].

Even if children are never explicitly told that mouses is the incorrect plural of mouse, they will naturally pick it up because of the lack of the words mouses in adult language, and the presence of mice [8]. This is because *experience* may be as powerful as direct feedback. Prediction allows us to expand our knowledge and beliefs about the world, but also enables us to compare against the experiences we have, using our experiences as a baseline

This idea is tied in closely with the belief that humans have an inner model of the world in our brain, which we are adjusting as we learn and grow. Nagai showed how children leverage this inner model to learn sensorimotor skills by comparing their predicted movements to the inputs from the real-world [9]. Winston called this inner model our inner language, which allows us to reason hierarchically and symbolically about the world without the burden of externalization [3]. Whichever way we chose to define it, we come back to the same idea: prediction is a key part of human cognitive development. We think that our ability to make predictions arises from our inner model of the world.

Story understanding, and Genesis specifically, naturally lends itself to this task. When Genesis processes stories, it translates them into an inner language called *genese*, which it then reasons about internally.[4] Because character alignment focuses on combining information from multiple sources, it will be able to take advantage of the same phenomena we saw with children: learning from experience. By building OPERA on top of Genesis, we enable the system to reason about the stories critically beyond the confines of the language structure.

2.2 Outcome Prediction in Systems Is Not New

The task of outcome prediction in intelligent systems is not new. The idea was first visited in the 1970s and 80s when there was a lot of work being done in learning and reasoning by analogy. The basic idea was that systems could be given a new situation (S_{new}), and if the system had seen a similar enough situation previously (S_{prev}), then is could draw an analogy between S_{prev} and S_{new}, and use information about S_{prev} to make prediction about $S_{new.}$ As story understanding systems started arising in the early 2000s, the idea of story alignment came about. In story alignment, two stories are aligned in a manner similar to drawing an analogy between them. This alignment allows prediction to become a form of gap filling – events from one story can be transposed into the domain of the other story to fill in missing details.

There were also approaches to outcome prediction using conceptual dependency maps and recipes, where the system used the dependencies as a form of common sense for prediction [10]. Advances in machine learning and NLP starting around 2010 have shifted prediction into a statistical task. NLP methods increasingly focused on pattern matching/transfer, using techniques like n-gram frequency, plot consistency, and sentiment analysis as the basis for comparison.

[4] See Appendix A for details on Genesis

Machine learning saw an explosion of classification methods and neural networks, most commonly for selecting the appropriate ending given an input story. Recurrent neural networks specifically have been popular in the predictive task because they allow information to be built up and adjusted based on prior information.

While each of these techniques have advanced our understanding of prediction and improved intelligent system's predictive ability, there are still many open challenges. Current state-of-the-art models are unable to match human-level performance on simple tasks such as selecting the appropriate ending to a four sentence story [11]. We believe that in order to achieve human-level performance, we must build a system that models human predictive capabilities. This led us to take a symbolic, analogical-reasoning-based approach to prediction, inspired by human story understanding and reasoning capabilities.

2.3 Selection vs. Generation

One important distinction in outcome prediction systems is outcome selection versus outcome generation. Although both are forms of prediction, they result in a variety of different systems. In outcome selection, systems are given a story and asked to pick the best ending from a set of specified choices. Selection is considered the easier of the two tasks because the system is given a finite set of possibilities, reducing the solution space significantly, and avoiding the issue of text generation. Outcome selection also opens the door for prediction to become a classification/pattern matching task. While common NLP techniques like word frequency, n-grams, and keywords enable systems to learn statistical correlations between a story and the outcome and consequently achieve high performance, we believe these techniques do not accurately model human understanding and comprehension.

To highlight this problem, a group of researchers proposed the Story Cloze Test in 2017 [12]. The test served as an evaluation framework for prediction systems by focusing on causal and temporal commonsense relations between events. The test includes a database of four-sentence stories from everyday life matched with two plausible endings. The task was trivial for people (100% accuracy) but all the state-of-the-art methods achieved no better than random performance (approx. 51% accuracy) while humans achieved perfect accuracy. This test resulted in new approaches to prediction that incorporated techniques such as sentiment analysis and commonsense knowledge bases. However, many of these new approaches seem to be skewing farther towards the direction of applying all the techniques in hopes of achieving increased performance, rather than developing systems that better model human understanding and approaches to prediction.

The second type of task is outcome generation. This task is much harder because the system must generate the ending; there is substantially less work on this task. We believe outcome generation shows greater understanding because the system must process the story, apply past experience, and have a language model that allows it to generate new text that makes sense in the current situation. Because we built OPERA as a module on top of Genesis, we have extensive language modeling at our disposal. This allowed us to focus more on the prediction aspect of the challenge, then leverage the Genesis implementation for generating the outcome text.

2.4 Current Approaches

There have been a variety of approaches to the problem of outcome prediction, which can be roughly split into two categories: symbolic approaches and statistical machine learning methods.

We provide a brief account of current statistical machine learning approaches, leaving the bulk of this section to review symbolic approaches.

Most statistical machine learning approaches to prediction are based in deep learning and are focused in the story outcome selection task. Deep learning has brought about what many call the third wave of NLP [13] because it extracts features automatically and uses hidden layers to build up concepts in a hierarchical fashion. Neural network approaches typically use word2vec [14] or similar algorithms to generate word embeddings for text before sending it to neural networks. Most deep networks are built either as CNNs or RNNs.

CNNs are highly effective at classification tasks and have been used increasingly for sentiment analysis models. State of the art sentiment models are achieving over 90% accuracy on sentiment classification [15]. The authors of the story cloze test showed that although sentiment analysis could improve the accuracy of the submitted models (to about 60%), the systems still failed on most of the harder common-sense predictive tasks in the test. CNNs also have the drawback that they are very data intensive and cannot model the long-distance contextual information needed for prediction. RNNs are an alternative to CNNs and are better suited to prediction because of their recurrent nature, which inheritably lends better to sequence modelling (i.e., language). One of the recently popular methods includes Sequence to Sequence (Seq2Seq) learning, which is comprised of an encoder-decoder network built from LSTMs. Given a source sequence $(x_1, \dots, x_n)$ and a target sequence $(y_1, \dots, y_n)$, the goal of Seq2Seq learning is to learn the conditional probability of the target given the source. It does so by repeatedly encoding and decoding different parts of the source in an effort to learn the relationship between it and the target [16]. This approach is particularly useful for prediction because an outcome is usually based on the entire story i.e., not on a specific set of keywords or individual sentences. By analyzing the conditional probability of an outcome given the story sentences, Seq2Seq is able to better capture this relationship.

Most recent approaches to story prediction since the story cloze test are combining sentiment analysis, commonsense databases, deep neural networks, and generative modeling. One example is Guan et al.'s work on encoding commonsense for story generation [17]. Their work claim context clues and common sense are the key to generating an outcome that is reasonable and expected. They use an incremental encoding scheme built from LSTMs to capture the context clues (author's hints about the words or situation) in the story. They believe context clues are important for outcome generation because they capture how sentences relate to each other; usually context clues are spread across nearby sentences, where a given sentence may contain a context clue for the preceding or following sentence. They combine this encoding scheme with Concept Net [18] to get commonsense relations about what is happing within the story. Both are combined to generate a final one sentence output. They evaluated their system manually (through Amazon Mechanical Turk) for logic and achieved 50% accuracy, which compares favorably to methods like Seq2Seq – 25%.

We believe this shows that moving towards more human-like approaches to these tasks – such as using common sense and looking at context clues – will lead to better results. We have decided to take a symbolic approach to the prediction task, focusing on modelling human understanding and processing via story understanding systems. The remainder of this section reviews a couple of symbolic implementations and discusses how they align with this thesis.

Chaturvedi et al worked on overall story comprehension as a way of predicting story endings [19]. Their system was built in response to the Story Cloze test and is designed to take in a set of short stories and pick the potential ending from a set of two candidates. This results in the final decision being made by a classifier, but the bulk of their work is in more traditional semantic approaches. They look at 3 aspects of a story:

1. The sequence of events described in the story: the chosen ending should make sense

given what was happening prior (i.e., *Bob punches Sally* is much more likely to be followed by *Sally got upset* than *Sally started laughing*).

2. Its emotional trajectory: the ending should follow the same emotional arc of the main story.
3. Plot consistency: stories typically do not introduce new information at the end.

They weight these 3 aspects for each story and have a probability model that combines them to select an output. We have taken a similar approach, defining a set of characteristics by which we evaluate each character in a story. These characteristics will also be weighted and combined to produce a final output. However, our system is focused on outcome generation rather than selection and will be working from stories that are much longer and more complex (i.e., closer to 60 sentences rather than 4).

Fay took a different lens on prediction in his work *Enabling Imagination through Story Alignment* [4]. Also built on top of the Genesis Story Understanding System, his work focused more heavily on the optimizations needed to run story alignments with reasonable performance. His alignment algorithm, inspired by the Needleman-Wunsch algorithm for aligning DNA pairs, creates a similarity matrix between two stories with a binding list to ensure continuity of entities. He used heuristics to generate his binding list as a tree and was able to reduce the potential search space for alignments significantly. Fay does generation rather that selection, but his work restricts the story alignment task to only 2 stories. This means the results can be unimpressive if it is not a good alignment because his system has no other information. We build on the algorithm Fay designed, taking advantage of the many optimizations he proposes as we expand it to the larger scope of comparing multiple story sequences and predicting the result given multiple alignments.

Chen et all [20] also attempted the Story Cloze Test by combining narrative sequence, sentiment analysis, and common-sense knowledge. Their system was built from three modules –

an encoder/decoder framework to extract sentiment, an LSTM to model narrative sequence, and a Concept Net expert – that were combined to train the neural network. The most interesting outcome of their work was their analysis of performance provided by the 3 modules they used. Using all 3 modules, their network achieved approximately an 87% success rate. They then showed that if they trained their network with narrative sequence alone, it would obtain almost 85% based on that alone. Removing narrative sequence from the model but keeping the other two reduces performance to 65%. We believe this supports the hypothesis that story understanding a sequence of events is really at the core of how we understand and reason, and therefore contributes disproportionately to the prediction task. Common sense and sentiment analysis alone are not enough because you need to be able to see how one event leads to the next.

3 Character Alignment as an Approach

In this chapter we present character alignment as an approach for outcome prediction in story understanding systems. Character alignment is inspired by work on story alignment and borrows heavily from ideas in analogical reasoning. The goal of the character alignment approach is to predict the outcome of a certain character $\mathbf{C_A}$ in the current story by drawing analogies to characters in past stories $\mathbf{C_P}$s. We describe the conceptual framework of our approach in the style of Hall's analogical reasoning framework:

1. **Recognition** of *relevant* past characters ($\mathbf{C_P}$) from stories
2. **Elaboration** of the analogical alignment between each $\mathbf{C_P}$ and $\mathbf{C_A}$
3. **Evaluation** of the alignment
4. **Consolidation** of the outcome

A detailed walkthrough of the implementation of this approach can be found in chapter 4.

3.1 Analogical Reasoning as an Inspiration

Humans use analogies every day to compare and apply prior experiences. We do this unconsciously, such as when we walk into a room and think "*this reminds me of xx"* or when we meet a new person and we think *"they remind me of yy"*. In all these cases, we are drawing invisible connections between prior experiences and our current situation, which guides our predictions.

Formally, analogical reasoning begins with a set of questions that need to be answered about a situation:

Have I seen this situation before?

Have I seen the same situation in a slightly different form?

Do I know related situations?

To answer these questions, computational models need to have a <u>representation</u> that encapsulates the situation, an <u>algorithm</u> that can compare these situations to each other, and a <u>measure</u> of similarity. This provides the basis that we can build on to create analogical mappings that can be used to infer information from one situation to another. In the late 80s, when research on analogical reasoning was at its peak, Hall provided a comprehensive review of computational approaches to analogical reasoning and analyzed the goals of the research, which spread across a multitude of fields [21]. We highlight two of the approaches below that impacted our work.

In the late 60s and early 70s, Becker was already proposing analogies to make predictions via simple schemas [22]. His schemas were made up of simple *facts* and *rules* that related facts to each other. He proposed a weighting scheme based on the idea of the **criticality** between two structures A and B – a measure of the degree to which the presence of B in A is responsible for the distinctive identity of A. This is especially important in analogy where the system needs to know what to pay attention to. An example of a fireman rule is in figure 3.1.1 below:

$$R_1: \left[\{\langle \mathtt{member}^4\ \mathtt{Rupert}^2\ \mathtt{fireman}^4 \rangle :4\} \Rightarrow \left\{ \begin{array}{l} \langle \mathtt{wears}^4\ \mathtt{Rupert}^2\ \mathtt{CC}^2 \rangle :4 \\ \langle \mathtt{member}^4\ \mathtt{CC}^2\ \mathtt{suspenders}^4 \rangle :3 \\ \langle \mathtt{property}^4\ \mathtt{CC}^2\ \mathtt{red}^4 \rangle :3 \\ \langle \mathtt{dances\text{-}with}^4\ \mathtt{Rupert}^2\ \mathtt{Maude}^4 \rangle :1 \end{array} \right\} \right]$$

Figure 3.1.1: Example of one of Becker's rules. The fact on the left is related to the set of facts on the right, forming the rule. This rule provides a way to identify fireman (specifically Rupert) through the fact that he is wearing red suspender and dances with Maude. Figure sourced from [22]).

Becker's proposed system would theoretically use these rules to make predictions in novel situations (i.e., seeing another person in red suspenders may imply they are a firemen). Although he provided a relatively complete model for his schema idea, most of it was never implemented.

His work represented one of the early attempts at prediction via analogy, and although his theoretical situations were short and simple, they were a first step toward accumulating and applying past knowledge.

In the 1980s, Winston was researching analogies between characters for the goal of learning and reasoning [23]. His work identified relations in stories which he used to measure how connected two storylines are, e.g. Romeo and Juliet compared to Prince Charming and Cinderella. Like Becker, Winston applied a weighting scheme to prioritize the causal relations that were most important to generate a match score. He demonstrated how reasoning by analogy requires attention to a lot of details, including how the characters relate to each other in terms of plot, actions, and inherent properties. He achieved this measure of relatedness through a primitive version of what is now WordNet [24], where a system can capture the hierarchical nature of words. An example of this hierarchy can be seen in figure 3.1.2 below:

```
   Romeo is a type of boy, that is a type of man,
that is a type of person.

   Charming is a type of prince, that is a type of
man, that is a type of person.
```

Figure 3.1.2: Example of word hierarchy with Romeo and Charming. We can go up the hierarchy from Romeo → boy → man → person and Charming → prince → man → person.

It is important for the system to understand that Romeo and Charming are both persons (specifically men). In some situations, the level of specificity (boy and prince) may be important but in others it may not, and we want to account for both. The key is that the more you know about how two situations relate to each other, even at a more general level than they are described, the better you can use them later when drawing analogies. In short, systems needed to have commonsense knowledge about the world they were analyzing. Winston achieved this through these abstractions, Schank [10] achieved this through his scripts, and today's databases like

WordNet and ConceptNet [18] provide a framework through which to instill commonsense knowledge into programs.

We believe analogical reasoning can be combined with recent advances in story understanding and NLP, enabling analogical reasoning methods to be more powerful. One example of this is our character alignment method, one of the main contributions of this work.

3.2 Character Alignment

Character alignment is an approach by which story understanding systems can apply past experiences to make predictions about a character's outcome. It is based heavily on ideas of analogical reasoning and provides an alternative to traditional story alignment methods and to newer neural network techniques. Character alignment works in four steps:

a) **Create** a representation of each character
b) **Recognize** potential alignments between past characters C_P and a specified character C_A
c) **Evaluate** the alignments quantitatively
d) **Predict** the outcome of C_A given the alignments

We use the following extremely abbreviated version of *Macbeth* with the ending removed (represented by the bolded sentence) as a motivating example:

> *Duncan is the king. Macbeth wants to become king. The witches give a prophecy. Macbeth stabs Duncan. Lady Macbeth kills herself. Macduff flees. Macbeth murders Lady Macduff. The prophecy comes true. Macduff confronts Macbeth.* ***Macduff kills Macbeth****.*

As we explain each step of character alignment, we will walk through a general example in which we are predicting an outcome for the character Macbeth by aligning it to the characters Hamlet and Claudius from *Hamlet.*

3.2.1 Action-Oriented Representations Capture Behavior

The approach begins by creating an action-oriented representation of each character that encapsulates their behavior in the story. The representation for each character is built from 3 core parts:

1. The plot units a character is involved in (*e.g. "Macbeth stabs Duncan", "Macbeth murders Lady Macduff").*
2. The concepts a character is involved in *(e.g. Revenge, Answered Prayer).*
3. The character's desires *(e.g. "Macbeth wants to become king").*

We argue that each of these contributes important information about a character's behavior.

Plot Units Provide a Comparative Baseline

The plot units[5] make up the first part of the representation and are important because they give a <u>temporal</u> sequence of a character's actions and interactions. The temporal sequence maintains the natural plot order while the system tries to align the greatest number of plot units. It is important that this alignment is not specific (i.e., not matching word for word), but instead uses a lexical database like WordNet and some way of integrating commonsense reasoning to allow for more generalized matching.

We start by drawing a distinction between story alignment and character alignment. In traditional story alignment methods, the entire plot of both stories is used to make comparisons. Character alignment instead focuses only on the parts of the plot that involves characters; any other

---

[5] See definition in footnote on page 15

parts are discarded (e.g. “The prophecy comes true”). However, plot alone is not enough. It is not enough to know what character does, there needs to be sense of understanding of motivations. This is where concepts and desires come in.

Concepts and Desires Answer the “Why” Question

We refer to c*oncepts* as they are defined in the Genesis system [2]: high-level themes/tropes that are instantiated from a specific set of phrases/actions in a story. They provide an abstraction from the details of a story to common patterns which can be used to identify commonalities between stories. An example is provided in figure 3.2.1 below:

```
Revenge:
XX is a person. YY is a person
XX harming YY leads to YY harming XX
```

Figure 3.2.1: Revenge concept. Concepts are predefined in Genesis and are used to represent broader themes/tropes in a story. More information can be found in appendix A.2

Concepts allow us to abstract a level up from the plot and analyze how the situations relate to each other. They also give perspective to a given situation. Consider the full version of the *Macbeth* short story without the ending removed and the subsequent question:

> *Duncan is the king. Macbeth wants to become king. The witches give a prophecy. Macbeth stabs Duncan. Lady Macbeth kills herself. Macduff flees. Macbeth murders Lady Macduff. The prophecy comes true. Macduff fights Macbeth. Macduff kills Macbeth.*
>
> **Question:** Why does Macduff kill Macbeth?

Although there is no explicit explanation in the story as to why Macduff kills Macbeth, the most likely answer is "because Macbeth killed Lady Macduff". This answer comes from the connection we implicitly draw between Lady Macduff and Macduff. Given this connection, we infer that if Macbeth murders Lady Macduff, this will probably anger/upset Macduff, who may then avenge her death by killing Macbeth. This reasoning could be represented by a concept such as:

```
                    Avenge Family

                 yy is zz's relation.
        xx's harming yy leads to zz's harming xx.
```

The concept gives the system more information about *why* Macduff did what he did. This is especially important when aligning characters because there is a difference between a character that harms another character out of fun/enjoyment versus one that does so to avenge a family member. Concepts allow refinement of comparison.

Desires further this refinement by focusing on situations in which a character *wants* something. Desires are important because they speak directly to a character's motivation towards a certain action. In *Macbeth,* the fact that "Macbeth wants to be king" is crucial information because it later explains *why* Macbeth stabs Duncan (who is the king).

As with the plot units, both concepts and desires help quantify how similar characters are to each other. When you put all 3 together you get a fuller picture: *what* did the character do, what did the character *want* to do, and what are the higher level *concepts* the character was involved in.

3.2.2 Alignment Happens at Two Levels

The action-oriented representations we built encapsulate each character within the situation of their story and are the input to the alignment algorithm. We perform two levels of alignment: a primary alignment to match the previously seen characters, $\mathbf{C_P}$s, to the current character, $\mathbf{C_A}$, and then a secondary alignment to match the $\mathbf{C_P}$s to each other. The primary alignment is the matching step common to many analogical reasoning implementations, but it does not get us all the information we want. To see this, let us revisit the questions at the beginning of this chapter:

Have I seen the situation before?

Have I seen the same situation in a slightly different form?

Do I know similar situations?

In our case, the *situation* is represented by our character representations. With primary alignment, the characters that are aligned via their plot, concepts, and desires with the original character, answer the first two questions. However, the last question - *do I know related situations* - is more difficult because it is asking for generalization. It is not simply asking you to identify a *single* situation but instead is asking you to reason about *multiple* situations. It requires understanding how the situations relate to each other, using that to reason about how they might collectively relate to a given situation. The secondary alignment is our solution to the final question, by moving beyond a single alignment to group alignment that allows for the recognition of patterns among multiple other characters seen in the past. We want to repeat the same process we did in the primary alignment of $\mathbf{C_A}$ to $\mathbf{C_P}$ within the set of $\mathbf{C_P}$s themselves to get a sense of how they relate to each other. This will create groupings of related characters that can then be reasoned about together in relation to $\mathbf{C_A}$.

In the context of our Macbeth example, primary alignment would be aligning Hamlet to Macbeth and aligning Claudius to Macbeth, and secondary alignment would be aligning Hamlet and Claudius to each other. This could result in Hamlet and Claudius being grouped together and then compared as a group to Macbeth.

3.2.3 Evaluating the Alignments and Predicting the Outcome

At this point, the character alignment algorithm has the following information

- An **action-oriented character representation,** for each character in the stories read previously (*e.g. Macbeth, Hamlet, and Claudius).*
- A set of **primary matches, *M*,** that contains all the characters $\boldsymbol{C_p}$ that matched to $\boldsymbol{C_A}$ (*e.g. Hamlet-Macbeth; Claudius-Macbeth*).
- A set of **secondary matches,** that records how characters in ***M*** match each other. (*e.g. Hamlet-Claudius*)

To distinguish which alignments are useful in making a prediction about $\boldsymbol{C_A}$, we need a method for scoring them.

The weighting scheme lets us focus on what is important

The goal with a scoring system is to focus on the information that is most relevant to the analogy. We do this by applying a weighting scheme across all the matches in which the concepts are given the most weight, followed by desires, and last the plot. We do this because concepts encapsulate the actions that a character takes and therefore speak the most to their behavior. Desires are next because they represent what a character wants and can be especially important in evaluating what they might do in a certain situation. Plot is weighted the lowest because characters may have a lot of plot overlap even if they are not similar otherwise just because of some of the interactions they have. This scheme forms the building blocks of the overall scoring system. As the matches are combined into groups and interim predictions made, the weights are combined, enabling a consistent measure of confidence throughout the system for each part.

In the context of the Macbeth example, this would result in assigning the primary match *Hamlet-Macbeth* a score of $\mathbf{s_1}$, the primary match *Claudius-Macbeth* a score of $\mathbf{s_2}$, and the secondary match *Hamlet-Claudius* a score that is some **function of $\mathbf{s_1}$ and $\mathbf{s_2}$**. One area of future work is defining a dynamic weighting scheme that updates based on the stories read and the importance of each concept/desire to the story. These ideas are discussed further in the chapter 6.

Partial concepts give us a glimpse into the future

In the previous section, we explained that concepts are important because they represent common themes/tropes. However, sometimes there exists a partial concept; we can see a known concept beginning to play itself out even though it has not fully developed. This is an important application of our prior experience because we know that this concept has played out with other characters in prior stories. Therefore, we want to match up those characters with full concepts to the corresponding partial concepts we find.

In the context of our Macbeth example, Avenge Family[6] is a partial concept for Macbeth because Macbeth has harmed Lady Macduff, who is Macduff's relation. We can link this partial concept to the match between Claudius and Macbeth because Claudius has a full Avenge Family concept: Claudius harming King Hamlet (Hamlet's father) leads to Hamlet harming Claudius.

There are a few things to make note of here. First, we avoid using partial concepts to find alignments unless there were no matches found using the concepts and desires, because we want the alignments to be a product of actions that have occurred. The partial concepts augment the

---

[6] Avenge Family: `yy is zz's relation. xx's harming yy leads to zz's harming xx`

match by using the $\mathbf{C_P}$'s concepts to give insight into the outcome of $\mathbf{C_A}$. It is also important to score the partial concept based on how much it has been instantiated in the story. Consider the concept crime of need:

```
Crime of need
XX is not a criminal
XX needs YY.
XX can't get YY.
XX commits a crime to get YY.
```

If the only part of the concept we have matched is `CA is not a criminal`, that partial match is not very useful because there is not much evidence to make us believe $\mathbf{C_A}$ will complete the rest of the concept. However, if $\mathbf{C_A}$ matches the first 3 statements, we may be much more likely to think that $\mathbf{C_A}$ may commit that crime in the future, finishing the concept[7]. Therefore, partial concepts are based on the existing matches and $\mathbf{C_A}$**'s** story.

The prediction combines multiple sources of information

The final step in the character alignment algorithm is producing the prediction. This happens in a pyramid-like fashion where multiple interim outcomes are proposed, and the systems chooses the outcome accordingly. The process begins with the groupings created during the secondary alignment. Each of the groupings will produce a prediction based on the shared characteristics of the group in relation to the main character $\mathbf{C_A}$, including how the plots overlap and the concepts and desires shared. Each of the partial concepts will also be evaluated and those with the highest score will produce an interim prediction as well. Each of these interim predictions will be weighted based on how similar the group of characters was to the original character $\mathbf{C_A}$ and combined to enable a final prediction to be generated. This process allows us to combine all the information

[7] See appendix A for further discussion on concepts matching/firing in story understanding

we've gathered about how characters align to each other – individually through their plot, concepts, desires, and partial concepts, and jointly as a group with their shared similarities – in order to make the final prediction.

In the context of our example, this final step would have the secondary alignment match *Hamlet-Claudius* generate an interim prediction as well as the avenge family partial concept. These predictions would be combined to generate the final prediction for Macbeth, which could be *Macduff harms Macbeth.* See chapter 4 and chapter 5.1 for more detailed examples.

In this chapter, we explained the conceptual framework of the character alignment approach. We reviewed prior work on analogical reasoning and showed how it inspired this approach. We detailed the approach by discussing how we choose to represent information (action-oriented character representation), how we find relevant past experiences (primary and secondary alignment), how we evaluate those past experiences (weighting scheme), and how we generate a prediction (combining multiple sources). In the next chapter, we will present an implementation of this approach.

4 OPERA: A Computational Approach

In order to fully specify and test our character alignment approach for prediction in story understanding systems, we designed and developed OPERA (**O**utcome **P**redication **E**nabled by cha**R**acter **A**lignment). OPERA is built as module on top of the Genesis system, which allows it to take advantage of Genesis's expansive story understanding abilities. The OPERA system implemented character alignment through the following four major steps (see figure 4):

1. **Story Processing:** Extracting information we believe to be relevant
2. **Remembering Information:** Storing what we consider to be necessary information from stories into graph-like data structure for reference later
3. **Character Alignment:** Aligning characters from previous stories to the character in the current story
4. **Outcome Prediction**: Generating a prediction for the current character based on the alignment

In order to better explain the OPERA implementation, we will be walking through the following example in this chapter, which we will be referring to as the **Shakespeare Example**: OPERA reads *Julius Caesar* and *Hamlet* as background and *Macbeth* as the main story with the ending removed, and is prompted to predict an outcome for the character Macbeth.[8]

4.1 OPERA Gains Experience by Processing Stories

In order to make informed predictions about a given character outcome, OPERA needs to have a sense of how characters behave and why. OPERA builds this by processing stories to create an action-oriented representation of each character that encapsulates their behavior in the given story.

[8] See appendix B for stories.

The OPERA system itself does not have a story processing engine; rather it sends the stories to Genesis to be read and Genesis provides the inner model representation. This begins OPERA's process of identifying the main characters in the story. For each identified character, a representation is built based on 3 core parts:

1. The plot units a character is involved in
2. The concepts a character is involved in
3. The character's desires.

OPERA uses the representation provided by Genesis to identify the parts of the plot specific to each character. Any part of the story that is not about a character is discarded. OPERA also flags all the story units that contain desires, which are processed separately. The concepts are extracted by the Genesis concept expert system and sent to OPERA. OPERA breaks down the concepts to identify all the characters involved in the concept and the roles they play (see section 4.3).

4.2 OPERA Remembers Information from Each Story

OPERA needs at least one story for experience before it can make any predictions, but it performs better the more experience it gains, i.e., the more stories it reads. As OPERA processes each story, it stores the information in a graph-like data structure we designed called StoryWeb. Story Web is at its core an undirected graph made up of 3 types of nodes and 3 types of edges; the different node types represent abstract ideas; the edges connect specific instances of those ideas to each other (see table 4.2 and figure 4.2a on the next page).

Nodes	Character Node	Represents a specific character
	Concept Node	Represents a concept
	Desire Node	Represents a desire
Edges	Concept Edge	Connects a CharacterNode to ConceptNode
	Desire Edge	Connects a CharacterNode to DesireNode
	Binding Edge	Connects 2 CharacterNodes

Table 4.2: The StoryWeb is a graph-like data structure. The different node types (character, concept, and desire) represent abstract ideas and the edges (concept, desire, and binding) connect specific instances of those ideas to each other.

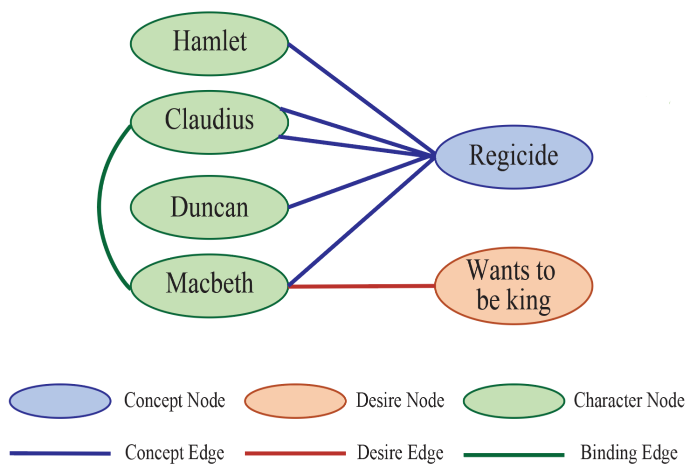


Figure 4.2a: Section of StoryWeb for the **Shakespeare example**. Each node can have multiple edges connected to it, such as Macbeth who has a concept edge to regicide, a desire edge to "wants to be king", and a binding edge to Claudius. If a character is involved in the same concept multiple times, then it will have multiple edges connecting it to that concept node (such as with Claudius connected twice to regicide).

The character node is the most important node in the Story Web because it is the starting point for all the edge connections in the web. All the edges are either connecting character nodes to each other or to instances of the other node types. As a result, OPERA can access all the useful information about a character by querying the character node. Concept nodes and desire nodes contain general information about the idea they are representing, such as the name and the rules that they are based on (see regicide node in figure 4.2b on the next page). They serve as a kickoff point for searches through the Story Web because they connect to every instantiation.

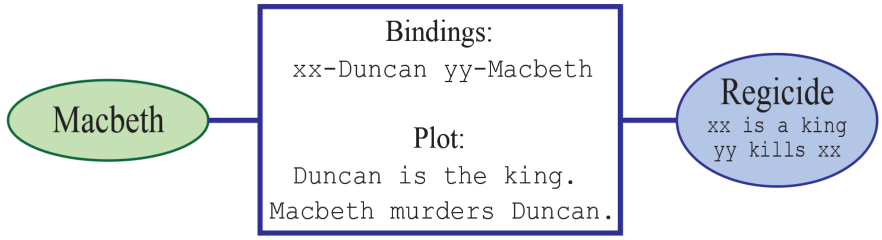


Figure 4.2b: Zoom in of regicide concept node and the concept edge between it and Macbeth. Each concept node contains the rules needed to instantiate it, which in the case of regicide is `xx is a king and yy kills xx`. The concept edge contains the specific instantiation information, which includes the parts of the story plot that were involved and the role bindings.

The instantiations are provided by the concept edges and desire edges. These edges not only serve a connective purpose between all the characters involved and the main node, but they also store information about the plot units that instantiated the edge and the bindings to each role in the edge (see figure 4.2b). This means each edge is tied to a single instantiation so if a character is involved in a concept multiple times, there will be a concept edge for each instantiation. The binding edge is unique because while all the other nodes and edges are created when the original story is being read, binding edges are created only when OPERA is performing character alignment and are thus dependent on the alignment OPERA is doing.

4.2.1 The StoryWeb Leads to Compactness

The main benefit of focusing on characters versus the entire story is compactness. OPERA can read in multiple stories for experience while being efficient about memory because it stores just a small amount of specific character information it needs about the characters and the story in the Story Web, and then discards the rest. This is a benefit over many other implementations, such as those that use story alignment, which require the entire story to reason about.

Consider the example of Macbeth. When Genesis reads Macbeth, the resulting story sequence contains **160 story elements**. Once OPERA processes this sequence, the resulting addition to Story Web is only **18 nodes** and **10 edges**. Of the 18 nodes, 8 of them are character nodes so OPERA is only creating 10 desire/concept nodes. Even setting aside the structure of Story Web which benefits faster search, processing 18 nodes and 10 edges will take vastly less time than processing 160 story elements.

In the example above, it is also important to notice that the number of nodes added is given the fact that OPERA has not read any other stories beforehand. If it had, the number of nodes would likely be even lower because Story Web is designed such that the nodes represent abstract ideas while the edges are specific instances. This means that each time a new story is read, OPERA adds only a limited number of nodes to the web (besides the character nodes), because many of the concept and desire nodes may exist already. We can see this in figure 4.2.1 below:

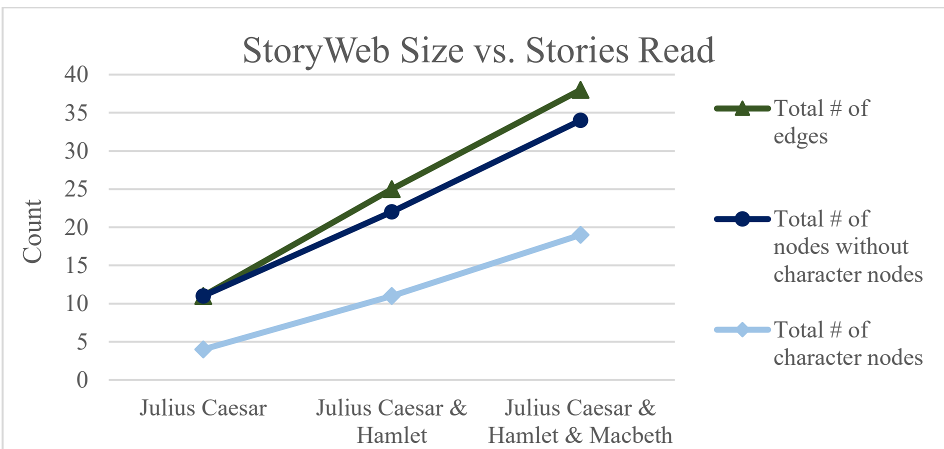


Figure 4.2.1: Graph of StoryWeb size vs. stories read for **Shakespeare example**. As OPERA reads in each story, it adds the information to the web. Note how the design of the StoryWeb data structure keeps it compact, with the number of nodes and edges growing roughly linearly.

This figure shows a graph of the size of the StoryWeb as OPERA reads the stories in the **Shakespeare example:** *Julius Caesar, Hamlet,* and *Macbeth* with ending removed. We chose to split the character nodes off from the concept and desire nodes, because the number of character nodes is simply based on the number of character in the story, and isn't as important when looking at compactness (since the character nodes are not searched over during alignment). We can see that the bulk of the information is added in terms of the edges connecting nodes, as this is the number that grows fastest. Note it grows roughly linearly, which is good since OPERA is designed to read in multiple stories at a time to gain experience. You can also see that the number of nodes that are added grows at a significantly smaller rater. This keeps the Story Web itself even more compact and makes it easier to search over in the alignment steps.

4.3 OPERA Aligns Characters Across Stories

When the user gives OPERA a character from the current story, C_A, and requests a prediction about the character's outcome, the first thing OPERA does is align that character to other characters, C_P, it has read about in any previous story. This is achieved in six steps (see figure 4.3 on the next page)

1. **Search:** Search for all C_P that align in some way with C_A.
2. **Node Alignment:** Match up the concepts and desires that are shared by each $C_P - C_A$ pairing.
3. **Partial Concept Matching:** Identify all C_A's partial concepts and match them up to each $C_P - C_A$ pairing.
4. **Plot Alignment:** Quantify the level of plot alignment between each $C_P - C_A$ pairing.
5. **Primary Character Alignment:** Combine the node alignment, plot alignment, and partial concepts to get the primary character alignment for each $C_P - C_A$ pairing.
6. **Secondary Character Alignment:** Align the C_Ps to each other to find groupings.

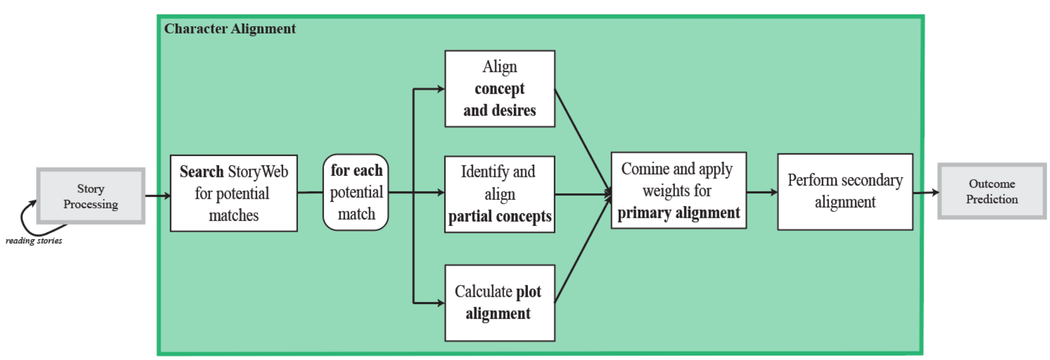


Figure 4.3**:** The character alignment algorithm uses the StoryWeb to identify potential matches and aligns each potential match to the main character via their concepts, desires, plot, and partial concepts (primary alignment). The matching characters are matched to each other and grouped in the secondary alignment step.

At the end of the alignment phase, OPERA will have created binding edges representing the connections that were found during the character alignment. These binding edges connect character nodes to each other and contain the information about how they are connected – such as how their plots overlap, and how their characters are similar – as well as a similarity score.

4.3.1 OPERA Searches for Potential Matches

For potential matches, the starting character C_A is the kickoff point. OPERA looks for all other character nodes that share some similarity with C_A by accumulating a list of all the concept/desire notes that C_A is linked to. From there, OPERA can use the connecting edges to get to the matching character nodes. Consider the example in figure 4.3.1a; Starting from the Macbeth node, OPERA moves into the regicide concept node, then fans out across the concept edges to look for matching characters (Hamlet, Claudius, Duncan, and King Hamlet). Each time OPERA finds a potential matching character C_p, it needs to validate the bindings i.e., determine whether the characters play the same role. OPERA does this by comparing the bindings contained in the edges to ensure a role match (see table 4.3.1a).

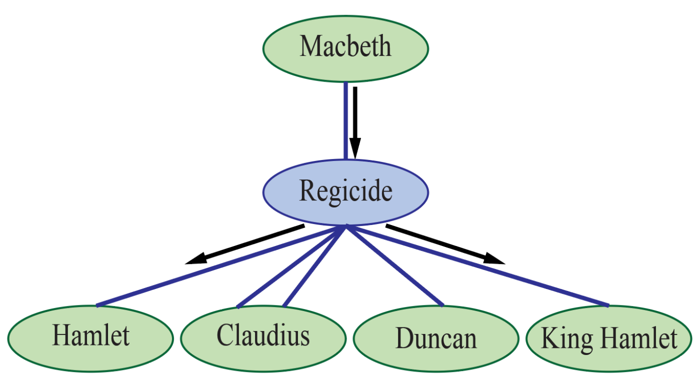


Figure 4.3.1a: Potential match search process in Shakespeare example. Starting at the Macbeth character node, OPERA can use the concept edge to cross into the regicide concept node, and from there can use any of the other concept edges to find other characters involved in revenge: Hamlet, Claudius, Duncan, or King Hamlet.

Regicide: XX is a king. YY murders XX. Macbeth binds to YY (i.e., Macbeth murders the king)		
Potential match C_p	Binding to Concept	Valid
King Hamlet	XX	No
Claudius	YY	**Yes**
Claudius	XX	No
Hamlet	YY	**Yes**
Duncan	XX	No

Table 4.3.1a: Binding validation results when searching for potential matches in Shakespeare example with regicide. Macbeth is bound to YY because Macbeth murders the king, so matching characters must also bind to YY in the concept. Claudius participates in regicide twice, playing different roles each time.

The first time C_A and C_p successfully match, a binding edge is created connecting them, containing a pointer to the edges they matched with. The next time C_p matches with C_A, the existing binding edge is expanded with the new information. Consider the example in figure 4.3.1b: the first time Macbeth and Claudius match, it is through the regicide concept node, so a binding edge is created between them with that information. The next time Macbeth and Claudius match, it is through the Answered Prayer concept node, so OPERA updates the existing edge with the new match information. At the end of the search step, C_A has a binding edge to each potential match that contains all the information on how they matched. The full results of OPERA's potential match search for the Shakespeare example is detailed in table 4.3.1c and 4.3.1d.

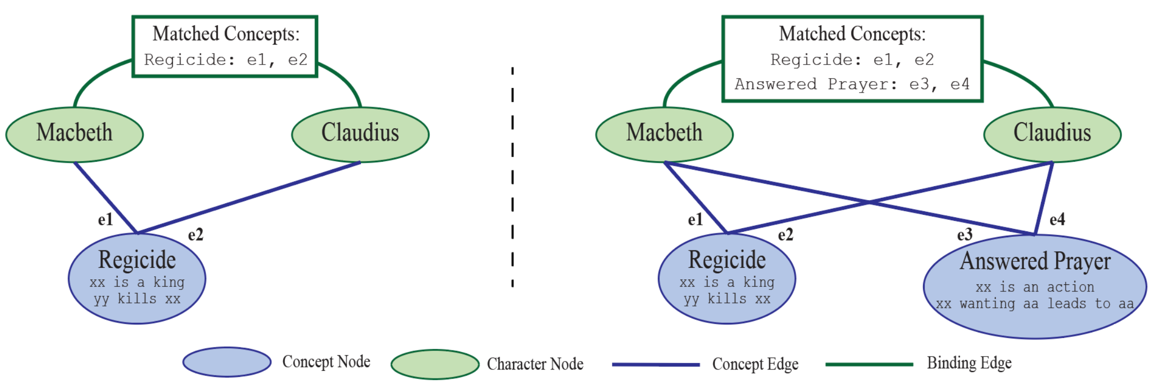


Figure 4.3.1b: Example of binding edge creation and expansion during matching process in Shakespeare example. On the left-hand side, we see the binding edge getting created when Macbeth matched to Claudius on the Regicide concept. On the right-hand side, we see that when Macbeth and Claudius match again, this time through the Answered Prayer concept, the binding edge is expanded to include the new information from that match.

Concept Name	Concept Description
Answered Prayer	XX wanting AA leads to AA
Mistake Because Harmed	XX makes a mistake by doing AA because XX doing AA leads to XX being harmed
Mistake Because Unhappy	XX makes a mistake by doing AA because XX doing AA leads to XX becoming unhappy
Regicide	XX kills YY and YY was the king
Revenge	XX harming YY leads to YY harming XX
Success	XX wanting AA leads to XX getting AA and leads to XX becoming happy

Table 4.3.1c: Concepts discovered in Shakespeare example with description. Formal descriptions and examples for all concepts can be found in Appendix C.

Binding Edge	Bound edges
`Macbeth—Cassius`	Mistake because harmed (CE)
	Mistake because unhappy (CE)
`Macbeth—Claudius`	Mistake because harmed (CE)
	Mistake because unhappy (CE)
	Success (CE)
	Answered Prayer (CE)
	Regicide (CE)
	Wants to be king (DE)
`Macbeth—Caesar`	Answered Prayer (CE)
`Macbeth—Hamlet`	Success (CE)
	Answered Prayer (CE)
	Regicide (CE)

Table 4.3.1d: Binding edges resulting from OPERA running primary alignment on Shakespeare example part 1: bound edges. Each binding edge keeps a pointer to all the edges via which the characters matched. (CE=Concept Edge; DE=Desire Edge).

4.3.2 OPERA Aligns the Matches Based on the Concepts, Desires, and Plot

Back at the original character node C_A, OPERA now has a binding edge to each potential match. In order to fully align the characters and maintain consistency during outcome prediction, OPERA needs to choose a single core binding set for each binding edge. In order to do this, OPERA searches for the binding set validates the greatest number of concept/desire edges. This is easily done because each edge contains the bindings that lead to that edge, so OEPRA simply has to search the edges and find a core binding set that is valid for the greatest number of edges (usually a superset of the edge bindings). Table 4.3.2 shows the core binding set chosen by OPERA for each binding edge in the Shakespeare example

Binding Edge	Bound edges	Core Binding Set
Macbeth—Cassius	Mistake because harmed (CE)	(Macbeth, Cassius) (Lady Macbeth, Anthony)
	Mistake because unhappy (CE)	
Macbeth—Claudius	Mistake because harmed (CE)	(Macbeth, Claudius) (Lady Macbeth, Hamlet) (Duncan, King Hamlet)
	Mistake because unhappy (CE)	
	Success (CE)	
	Answered Prayer (CE)	
	Regicide (CE)	
	Wants to be king (DE)	
Macbeth—Caesar	Answered Prayer (CE)	(Macbeth, Caesar)
Macbeth—Hamlet	Success (CE)	(Macbeth, Hamlet) (Duncan, Claudius)
	Answered Prayer (CE)	
	Regicide (CE)	

Table 4.3.2: Binding edges resulting from OPERA running primary alignment on Shakespeare example part 2: core binding set. The binding set enables OPERA to maintain character consistency during outcome prediction. (CE=Concept Edge; DE=Desire Edge).

Having a single set of bindings is important not just for the outcome prediction but also for plot alignment, where there is an expectation of continuity of characters. Fay discussed extensively how we expect that if two sequences of plot units are aligned with each other, then there should be character continuity i.e., the same characters remain bound to each other throughout the sequence [4]. This means that to align C_A and C_P's plots, we need to expand the core binding set into a full binding set containing all the other characters that appear in each of the stories.

To create this full binding set, OPERA starts with the core binding set and builds up the rest by testing all the possible binding combination for the remaining characters. OPERA leverages Fay's alignment algorithm to do this, which builds up these binding in a tree like structure, using the plot alignment as a heuristic for which binding combinations to explore further and which to discard. For example, consider the binding between Macbeth and Hamlet in table 4.3.2: the core binding set (`Macbeth-Hamlet;  Duncan-Claudius`) does not contain a binding for Lady

Macbeth. Therefore, OPERA needs to test every possible match to Lady Macbeth – e.g. `Lady Macbeth-Anthony, Lady Macbeth-Laertes`, etc. – until it finds the one that results in the greatest plot alignment, and so on for all the remaining characters.

Fay showed in his work how the computational effort of finding the best bindings can increase substantially as a sequence gets larger. He demonstrated how a brute force method can incur a cost of up to O(n!), where n is the number of plot entities to align. He detailed several optimizations that reduced the runtime to $O(n^2)$. By focusing on character plots rather than story plot, OPERA keeps the value of n significantly smaller since there are significantly fewer plot units to align. By starting with a core set of bindings, OPERA also reduces the computational effort because there is a baseline level of alignment that is built up rather than starting from scratch.

4.3.3 OPERA Augments the Matches with Partial Concepts

Partial concepts are useful for prediction because they give insight into what a character might do in the future. When OPERA is prompted to search for matches to C_A, it triggers the Partial Concept Expert (PCE) to run in the background. In 4.2, we explained how OPERA creates concept nodes to keep track of all the concepts it has seen as it reads. The PCE uses these concept nodes as a basis to search for partial concept instantiations in C_A's story. Each time it finds a partial concept instance, it records both the bindings and rules that lead to that instantiation as well as any missing/incomplete rules needed to complete it (see example in table 4.3.3a below). These missing/incomplete rules are used to quantify how complete the partial concepts are (compared to their complete counterparts) and are used to generate the interim predictions by OPERA later.

Revenge: XX harming YY leads to YY harming XX.	
Instantiated	XX harms YY: Macbeth harms Macduff.
Missing	YY harms XX Leads-to relation between XX harms YY and YY harms XX
Bindings	XX-Macbeth YY-Macduff

Table 4.3.3a: Example of a revenge partial concept for Macbeth. The concept was partially instantiated by the plot entity "Macbeth harms Macduff" and is missing "Macduff harms Macbeth" and a leads-to relation between them (see appendix A for further information on concept matching and leads-to relation). The bindings enable OPERA to fill in the next part of the partial concept during the outcome prediction step.

The partial concepts by themselves are not particularly useful however because we have no sense of whether C_A should complete them or not. This is where the binding edges become especially useful. OPERA augments each binding edge with all the partial concepts that C_p has as full concepts so that we can see which concepts C_A would be likely to complete based on its match to C_P. As we showed with the regicide concept in section 4.3.1, a character can also play different roles in a concept. This results in binding edges potentially having multiple instances of a partial concept, at varying levels of completeness, even if the full concept already existed in the edge. Table 4.3.3b shows the partial concepts identified and matched by OPERA for each binding edge in the Shakespeare example

At this point, OPERA has completed the binding edges. Each binding edge contains the matched concepts and desires, a full binding set, the aligned plots, and the matched partial concepts. With this, OPERA is now ready to evaluate the matches and complete the primary alignment.

Binding Edge	Bound edges	Partial Concepts and Completeness Percentage	
Macbeth—Cassius	Mistake because harmed (CE)	Revenge	71%
		Mistake because harmed	67%
	Mistake because unhappy (CE)	Mistake because unhappy	67%
		Answered Prayer	67%
Macbeth—Claudius	Mistake because harmed (CE)	Revenge	71%
	Mistake because unhappy (CE)	Mistake because unharmed	67%
	Success (CE)	Mistake because unhappy	67%
	Answered Prayer (CE)	Success	50%
	Regicide (CE)	Success	50%
		Success	33%
	Wants to be king (DE)	Answered Prayer	67%
Macbeth—Caesar	Answered Prayer (CE)	--	--
Macbeth—Hamlet	Success (CE)	Revenge	71%
		Revenge	71%
	Answered Prayer (CE)	Revenge	71%
		Revenge	71%
	Regicide (CE)	Success	50%
		Success	50%

Table 4.3.3b: Binding edges resulting from OPERA running primary alignment on Shakespeare example part 3: partial concepts. Macbeth's partial concepts are added to binding edges where the matching character has those partial concepts as full concepts. Next to each partial concept is the percent of completeness. Note some partial concepts are repeated multiple times, with varying levels of completeness; this is due to different roles/bindings in each. (CE=Concept Edge; DE=Desire Edge).

4.3.4 Primary Alignment Represents Direct Matches

For each binding edge, OPERA calculates a qualitative measure of how strong the match is between C_A and C_p using a weighted sum:

Type of match	Weight
Concept Match	3
Partial Concept Binding	3*(% completeness of partial concept)
Desire Match	1.5
Plot Match	1

Table 4.3.4a: Weighting scheme for matches. Concepts are weighted the highest, followed by desires and plot. Partial concepts are weighted based on how complete they are, and only the highest scoring instance of each partial concept is weighted and applied to the score.

These weights are currently predefined in the system. Concepts score the highest because they represent behavior rather than desires, which describe something a character wants but has no ties to whether it happens or not. The plot is weighted the lowest because characters may have a lot of plot overlap even if they are not similar otherwise, just because of some of the interactions they have. This is also why the plot alignment is not used as a criterion to find a match, but rather is performed after a match is established. Partial concepts also contribute to the score relative to how complete they are. If there are multiple instances of the same partial concept in a single edge, then the partial concept that is the most complete contributes to the score relative to their completeness. Table 4.3.4b shows how the binding edge score is calculated by OPERA for each binding edge in the Shakespeare example. At the end of the primary alignment, OPERA has a scored set of aligned characters connected to C_A.

4.3.5 Secondary Alignment Models Repeated Experience

The secondary character alignment aligns the aligned characters to each other and enables generalization of multiple experiences. Consider how humans make predictions: there are 2 common situations:

1. They have seen a similar situation once before, so they predict it will have a similar outcome.
2. They have seen a couple of similar situations before and they generalize an outcome based on those previous situations.

Primary character alignment is mainly situation 1: seeing how two character relate to each other in order to make a prediction. However, situation 2 is much more common and much more powerful: reasoning about multiple past events and combining them to make a generalized prediction. This is what secondary alignment aims to model. In the secondary alignment process, OPERA repeats the previous steps of character alignment, except for using the partial concepts instead of the complete concepts, on all the aligned characters, comparing them to each other. Table 4.3.5a shows the resulting binding edges from the secondary alignment process done by OPERA in the Shakespeare example.

Binding Edge	Bound edges and Weights	Partial Concepts and Score		Binding Edge Score
`Macbeth—Cassius` Plot Score: 4	Mistake because harmed (CE) – 3	Revenge - 71%	3*0.71 = 2.13	**18** 6 (bound edges) + 4 (plot alignment) + ~8 (partial concepts)
		Mistake because harmed - 67%	3*0.67 = 2.01	
	Mistake because unhappy (CE) – 3	Mistake because unhappy - 67%	3*0.67 = 2.01	
		Answered Prayer - 67%	3*0.67 = 2.01	
`Macbeth—Claudius` Plot Score: 10	Mistake because harmed (CE) – 3	Revenge - 71%	3*0.71 = 2.13	**36** 16.5 (bound edges) + 10 (plot alignment) + ~9.5 (partial concepts)
	Mistake because unhappy (CE) – 3	Mistake because unharmed - 67%	3*0.67 = 2.01	
		Mistake because unhappy - 67%	3*0.67 = 2.01	
	Success (CE) – 3	Success - 50%	3*0.50 = 1.50	
	Answered Prayer (CE) – 3	Success - 50%	--	
	Regicide (CE) – 3	Success - 33%	--	
	Wants to be king (DE) – 1.5	Answered Prayer - 67%	3*0.67 = 2.01	
`Macbeth—Caesar` Plot Score: 1	Answered Prayer (CE) – 3	--	--	**4** 3 (bound edges) + 1 (plot alignment) + 0 (partial concepts)
`Macbeth—Hamlet` Plot Score: 9	Success (CE) – 3	Revenge - 71%	3*0.71 = 2.13	**21.5** 9 (bound edges) + 9 (plot alignment) + ~3.5 (partial concepts)
		Revenge - 71%	--	
	Answered Prayer (CE) – 3	Revenge - 71%	--	
		Revenge - 71%	--	
	Regicide (CE) – 3	Success - 50%	3*0.50 = 1.50	
		Success - 50%	--	

Table 4.3.4b: Binding edges resulting from OPERA running primary alignment on Shakespeare example part 4: weighting results. OPERA calculates a score for each binding edge by performing a weighted sum across the bound edges, aligned plot, and partial concepts. Note that edges with multiple instances of the same partial concept, such as revenge in the `Macbeth—Hamlet` edge, only count the most complete partial concept in the score (ignored partial concepts have a -- in their cell). (CE=Concept Edge; DE=Desire Edge).

Secondary Binding Edge	Bound edges and Weights	Binding Edge Score
Cassius—Claudius Plot Score: 8	Mistake because harmed (CE) – 3 Mistake because unhappy (CE) – 3 Revenge (CE) – 3	**17** 9 (bound edges) + 8 (plot alignment)
Cassius—Hamlet Plot Score: 8	Answered Prayer (CE) – 3 Revenge (CE) – 3	**14** 6 (bound edges) + 8 (plot alignment)
Claudius—Caesar Plot Score: 2	Answered Prayer (CE) – 3	**5** 3 (bound edges) + 2 (plot alignment)
Claudius—Hamlet Plot Score: 9	Success (CE) – 3 Answered prayer (CE) – 3 Revenge (CE) – 3	**18** 9 (bound edges) + 9 (plot alignment)

Table 4.3.5a: Binding edges resulting from OPERA running secondary alignment on Shakespeare example. The four characters that matched to Macbeth in the primary alignment (Cassius, Claudius, Caesar, and Hamlet) are aligned to each other to quantify their similarity. Note that not all characters may match to each other in secondary alignment (such as Caesar who only matched to Claudius). (CE=Concept Edge; DE=Desire Edge).

There are a couple things to note about the differences between the primary and secondary alignment process. First, partial concepts are not used at all in the secondary alignment process because all the aligned characters are from stories that are complete, and therefore we know that the partial concepts will never be completed in those stories. Also, not all characters may match to each other in secondary alignment. Since we are comparing characters C_P that originally matched to C_A, there is no guarantee they will have anything in common with each other. We have found however that there is usually some commonality, especially with characters that matched strongly to C_A, because the same similarities they share with C_A are likely to be shared between each other.

Once all the aligned characters are aligned to each other, OPERA groups similar characters together. The groups are created based on how well the characters match each other. We want to group only characters that are well matched to each other, so each character has a dynamic threshold equal to 75% of its strongest match. OPERA looks through the binding edges from each character C_i to other characters C_j to find the C_j with the highest score, and then sets the threshold for C_i to be 75% of that. Any binding edges from C_i to C_j that come in below that threshold are removed. This process is repeated for each character until all the edges remaining pass the threshold hold test for each character. Table 4.3.5b continues the Shakespeare example with the thresholding process.

Secondary Binding Edge	Binding Edge Score	Threshold Value	Passes threshold
`Cassius—Claudius`	17	17*0.75 = 12.75	Yes
`Cassius—Hamlet`	14		Yes
`Hamlet—Cassius`	14	18*0.75 = 13.5	Yes
`Hamlet—Claudius`	18		Yes
`Claudius—Caesar`	5	18*0.75 = 13.5	**No**
`Claudius—Cassius`	17		Yes
`Claudius—Hamlet`	18		Yes

Table 4.3.5b: Threshold process for secondary alignment groups in Shakespeare example. The threshold value for each character is calculated by taking the highest scoring binding edge for that character and multiplying it by 0.75. Note both Hamlet and Claudius have the same threshold value because their highest scoring edge has a score of 18. All the binding edges are then tested against this threshold to ensure strong secondary matches.

From there, OPERA builds up the groups by iterating through the edges and building up cliques. Using cliques ensures that all characters in a group are strong matches to each other (i.e., if there are 3 characters in a group – C_i, C_j, and C_k – then there must be strong edges between all of them C_i-C_j, C_j-C_k, and C_i-C_k), Once this process is complete, each grouping contains between 1 (the singleton i.e., a character that didn't match well to anyone) and **N** characters, and no group is a subset of any other group. Each group is assigned 2 different scores:

1. **Match Score:** quantifies how strong a match the group (as a whole) is to C_A. Equals the average score of the binding edges between each character in the group and C_A.

$$Match\ Score = \frac{1}{n} \sum_{characters\ in\ group} binding\ edge\ (C_i - C_A)\ score$$

2. **Grouping Score:** quantifies how strong a match the characters are to each other. Equals the average score of all the binding edges in the group.

$$Grouping\ Score = \frac{1}{n} \sum_{character\ pairs\ in\ group} binding\ edge\big(C_i - C_j\big)\ score$$

Both these scores together play a role in generating the outcome. By quantifying how well the characters match to each other, the **grouping score** encapsulates OPERA's confidence in the group i.e., if the characters are not a strong match to each other, then the generalizations that OPERA makes about them as a group may not be as valuable. By quantifying how well the group matches to $C_{A,}$ the **match score** encapsulates OPERA's confidence in the analogy to C_A. Table 4.3.5c continues the Shakespeare example with the score calculations.

If there are no shared similarities among any of the characters, the results of the secondary alignment will be all singleton groups, which essentially mirrors the result of the primary alignment. However, if there are similarities, then characters will be grouped and the result will be fewer matches to C_A, but stronger matches overall. The groups created in the secondary alignment play the biggest role in producing outcomes.

Validated Binding Edges	Secondary Alignment Groups	Grouping Score	Match Score
Cassius–Claudius Cassius–Hamlet Hamlet–Cassius Hamlet–Claudius	Cassius Claudius Hamlet	**16.33** = 1/3*[17 (Cassius-Claudius) +14 (Cassius-Hamlet) +18 (Hamlet-Claudius)]	**25.16** = 1/3*[18 (Cassius-Macbeth) +36 (Claudius-Macbeth) + 21.5(Hamlet-Macbeth)]
Claudius–Cassius Claudius–Hamlet	Caesar	**0**	**4.0** (Caesar-Macbeth)

Table 4.3.5c: Scoring process for secondary alignment groups in Shakespeare example. We included the validated binding edges on the left-hand side for reference. The grouping score quantifies how well the characters in the group match to each other and is calculated from the validated binding edges (created during secondary alignment). The match score quantifies how well the group matches to the main character (in this case Macbeth) and is calculated from the binding edges created during primary alignment.

4.4 OPERA Makes a Prediction for a Given Character

The final step in the OPERA system is outcome prediction (see figure 4.4a). OPERA makes a prediction for C_A in a pyramid-like fashion, generating interim predictions from each of the secondary alignment groups and the partial concept instances that are then combined to produce a final prediction for C_A (see figure 4.4b) prediction for C_A.

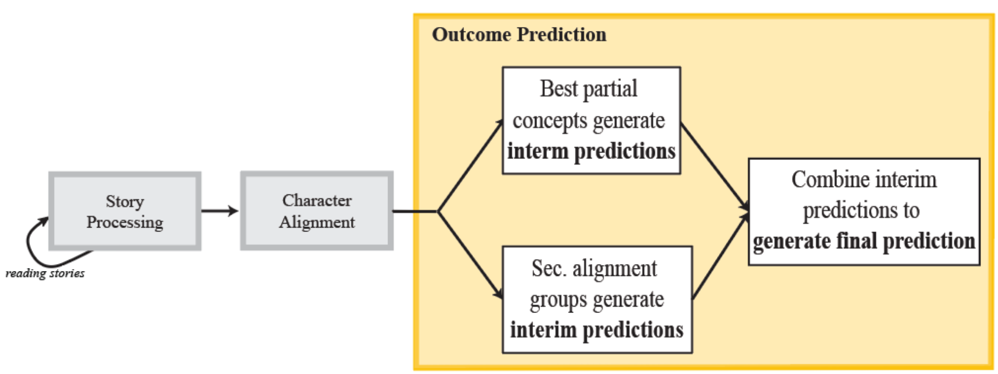


Figure 4.4a: To make a prediction, OPERA first searches for partial concepts in C_A's story. The partial concepts that are the most instantiated are selected to generate interim predictions about C_A. Each of the secondary alignment groupings also propose an interim prediction based on the plot alignment. The interim predictions are combined to generate the final prediction.

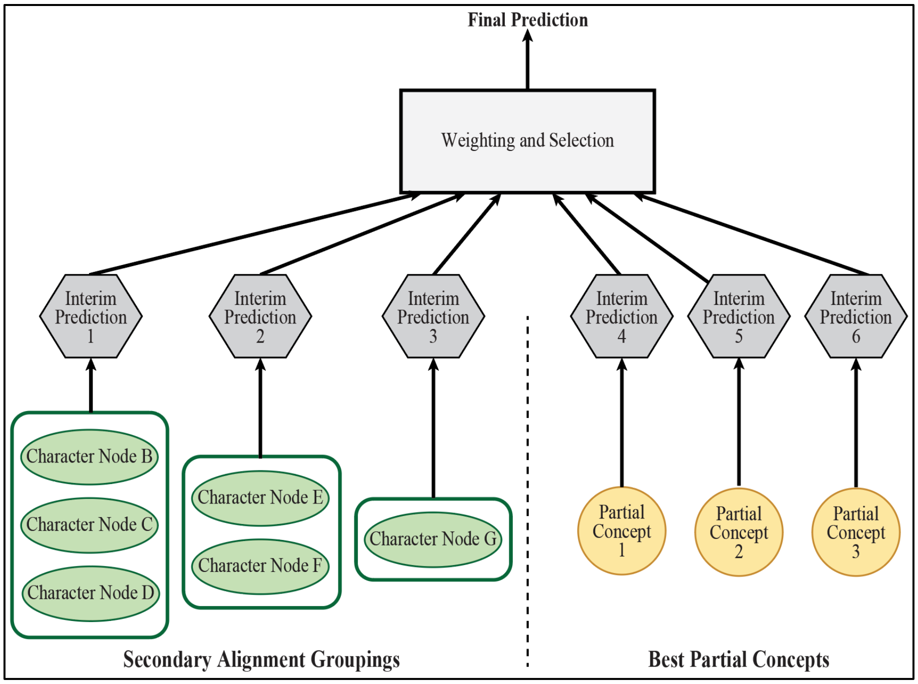


Figure 4.4b: Outcome prediction pyramid structure. Each of the secondary alignment groups and the best partial concept instances independently generate interim predictions. All these interim predictions are weighted based on how strongly the source (either the group or the partial concept) matched with $\mathbf{C_A}$, before they are combined to produce the final prediction.

4.4.1 Partial Concepts Give Hints about Future Actions

Partial concepts are important because they give insight into what a character might do in the future As explained in 4.3.3, OPERA augments each binding edge created in primary alignment with all the partial concepts that C_p has as full concepts so that we can see which concepts C_A would be likely to complete based on its match to C_P. The stronger the match between C_A and C_P, the more likely it is that C_A will complete a partial concept that C_P completed. Now in the outcome generation phase, OPERA looks at each binding edge and chooses the partial concept with the highest completeness percentage, using it to generates an interim outcome that would complete the next part of the concept. OPERA validates each interim outcome to ensure they are plausible (i.e., a character that has died cannot be involved) and that they do not already exist in the story. Once validated, each interim outcome is assigned a score that combines the score of the binding edge and the partial concept's completeness percentage. Table 4.4.1 shows the partial concept interim outcome generation for the Shakespeare example.

Binding Edge	Most-Complete Partial Concept	Generated Interim Outcome	Valid outcome?	Interim Outcome Score
Macbeth–Cassius	Revenge – 71%	Duncan's guards harm Macbeth.	No (Duncan's guards are dead)	--
Macbeth–Claudius	Revenge – 71%	Duncan's guards harm Macbeth	No (Duncan's guards are dead)	--
Macbeth–Caesar	--	--	--	--
Macbeth–Hamlet	Revenge 71%	Macduff harms Macbeth.	Yes	**15.27** 21.5 (binding edge score) *0.71 (partial concept score)

Table 4.4.1 Partial Concept interim outcome generation for Shakespeare example. The interim outcome is generated from the highest scoring partial concept for each binding edge, using the contained bindings. The valid interim outcomes are assigned a score determined by the binding edge score and the partial concept completeness

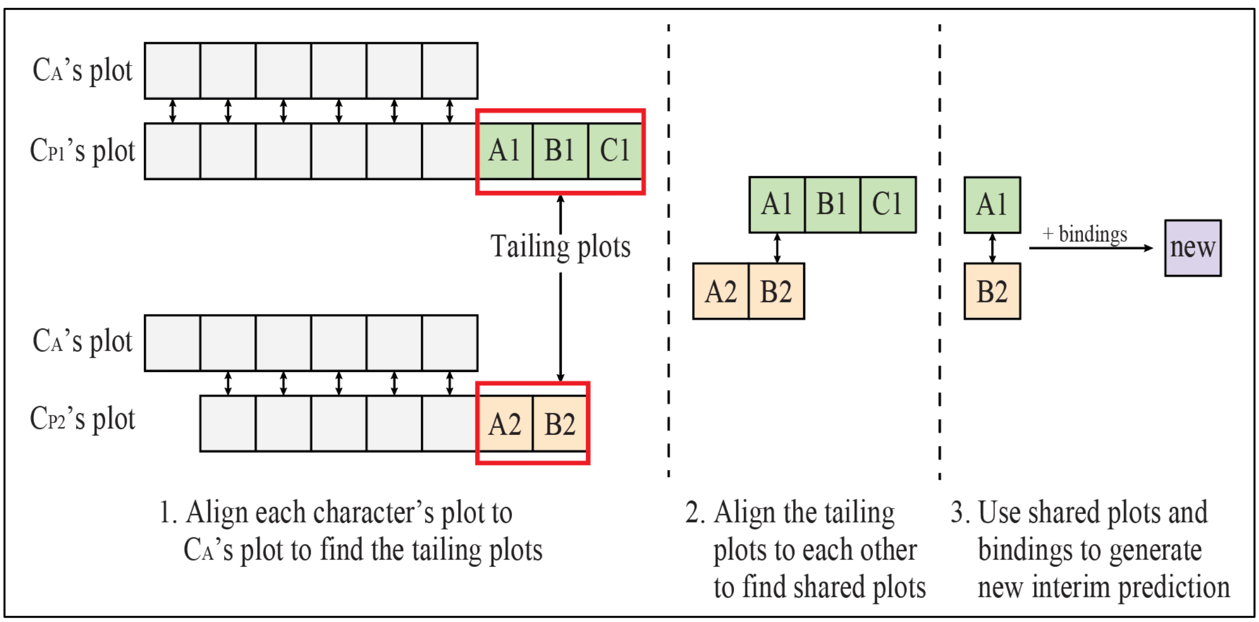


Figure 4.4.2: Process of generating interim outcomes using plot alignment in secondary alignment groups. Each character in the group has their plot aligned with C_A's plot to find the tailing plots (plot units remaining past C_A's plot, which are then aligned to each other. The shred plots are used to generate the interim prediction.

4.4.2 Secondary Alignments Produce Joint Predictions

Each secondary alignment group also produces an interim prediction. For each character in the group, OPERA uses the plot alignment between that character and C_A to extract the **tailing plot** – the remaining plot of C_p after the plot of C_A stops. All the tailing plots in the group are then compared to each other and reduced to the plot units that are shared by the most characters in the group (see figure 4.4.2). OPERA uses the bindings from the binding edge to generate an interim prediction such that all the characters are from C_A's story. Each prediction is assigned a score based on how strongly that group matched C_A.

4.4.3 Commonsense Rules Connect Interim Predictions to Produce an Outcome

To produce an outcome, OPERA needs to combine the interim predictions generated by the secondary alignment groups and the partial concepts (see figure 4.4b). The first part of this combination is condensing duplicate predictions which may occur when the same prediction is produced by the partial concepts and the second alignment groupings. The scores of these predictions are adjusted to account for the duplicate instances. Once there are no more duplicates, we check whether any of the remaining interim predictions have any commonsense connections to each other. Consider the example in figure 4.4.3 below.

```
Macbeth becomes king: 9
Macbeth becomes dead: 54
Macduff stabs Macbeth: 32
```

Figure 4.4.3: Interim predictions from OPERA with corresponding scores from Shakespeare example. Although *Macbeth becomes dead* has the highest score, commonsense tells us there may be a connection between that prediction and *Macduff stabs Macbeth* since people usually do not randomly become dead. OPERA leverages Genesis to look for these connections, using the connections to group the interim predictions together.

If OPERA picked the prediction with the highest score, it would pick *Macbeth becomes dead*. However, commonsense tells us that people do not just *become dead* out of nowhere – there is usually some casually connected explanation, such as an accident or a violent act. While OPERA does not have the ability to come up with possible explanations on its own, it can leverage the fact that it has multiple interim predictions that may contain the explanation to the act in question.

In order to avoid inadvertently drawing connections to improbable explanations, we first remove any predictions that score below a threshold, set to 20% of the highest scoring prediction. In the example in figure 4.4.3, this threshold would be set to 10.8 (i.e., 20% of 54), which would disqualify the prediction `Macbeth becomes king` with its score of 9. With the remaining predictions, we want to see if any of them are casually connected to the best prediction. To do this, we leverage Genesis and its commonsense rules and relations. We feed the remaining predictions into Genesis as if they were a story and extract all the connections Genesis draws between them.[9] The best interim prediction, along with any casually connected interim predictions, are combined into a single outcome. OPERA modifies the original story with this outcome and sends the modified story back to Genesis to generate the final version of the story. In the example in figure 4.4.3, Genesis draws a connection between `Macduff stabs Macbeth` and `Macbeth becomes dead`. OPERA then combines those predictions into a single outcome and sends it back to Genesis to generate our final version of the Macbeth story.

In this chapter, we presented OPERA as an application of character alignment. We explained how OPERA is built on top of the Genesis system. We described how OPERA processes the story sequences from Genesis to extract the concepts, desires, and plot for each character. We presented

[9] See Appendix A for elaboration on Genesis system.

the StoryWeb as the core data structure in the OPERA and showed how its implementation leads to compactness and enables search. We detailed how OPERA performs character alignment, including how it searches the StoryWeb for information, how it matches characters based on that information, and how it refines the matches to ensure consistency during prediction. We explained how OPERA performs two levels of alignment – primary and secondary – and how each of those alignments are used in generating interim predictions. Finally, we showed how OPERA uses commonsense rules and relations to combine interim predictions from the partial concepts and the secondary alignment groups to generate an outcome. In the next chapter, we show some examples of OPERA in action and further describe some of the optimizations implemented.

5 OPERA in Action

5.1 Experiments

We chose 5 stories to demonstrate and test OPERA's abilities – *Hamlet, Macbeth, Julius Caesar, The Lion King, and Anastasia*. Why only five stories, and why only these five? Is five enough and is there selection bias in choosing only these? The goal of our work was to demonstrate OPERA's capabilities, which is easily done with five stories. We believe the task of adding more stories was not needed in order to properly illustrate OPERA's use. Because of this, we also selected the stories so that they would illustrate interesting results. While they are similar to some degree, the themes found in these stories – revenge, success, mistakes etc. – are fundamental themes in human life and are common to many stories. Therefore, we believe that they serve as a good foundation to test a prediction system.

In our experiments, the order in which OPERA reads the stories is not important to the outcome prediction process; it matters only that one of the stories is incomplete. We use screenshots of Genesis's elaboration graphs for the story to show new connections and discoveries resulting from the predictions; more information on these graphs can be found in Appendix A.2.2 and in *The Genesis Enterprise* [2].

5.1.1 Experiment 1: OPERA's Prediction for *Macbeth* from Two Sides

In the first experiment, we will be looking at how OPERA is able to predict the correct ending to Macbeth. In this experiment, OPERA has read in *Anastasia, Lion King, Hamlet,* and *Julius Caesar* as background, and is currently reading *Macbeth* with the ending removed (see appendix

B5 for *Macbeth* with removed ending). Figure 5.1.1a shows the elaboration graph resulting from Genesis reading *Macbeth*. The StoryWeb breakdown prior to the outcome generation process is:

Total # of nodes = 56, total # of edges = 64.	
# of DESIRE edges = 15	# of CHARACTER nodes = 33
# of BINDING edges = 0	# of DESIRE nodes = 13
# of CONCEPT edges = 49	# of CONCEPT nodes = 10

Outcome Generation for Macbeth character

For reference, we have included following tables and figures in the next pages:

- Table 5.1.1a – the summarized results of the primary alignment for Macbeth
- Table 5.1.1b – the summarized results of the secondary alignment for Macbeth.
- Figure 5.1.1b – the elaboration graph of the shortened *Macbeth story* with the ending that OPERA proposes for Macbeth.

The primary alignment process results in 7 potential matches for Macbeth – Rasputin, Claudius, Scar, Hamlet, Cassius, Caesar, and Simba (see table 5.1.1a). One interesting thing to note here is strength of the matches (given in the binding edge score). In *Macbeth*, Macbeth is considered a "bad" character, and we can see that OPERA most closely matched Macbeth with other "bad" characters – Rasputin, Claudius, Scar, and Hamlet. This is a product of maintaining the consistency in the concept alignments because similar characters generally play similar roles in their concepts. We also note matches can have high scores both because of the number of bound edges and the number of different partial concepts. We can see this by comparing the [Macbeth-Rasputin] match to the [Macbeth-Claudius)] match. The both score pretty similarly (within 6 points of each other) but the [Macbeth-Claudius] match has more bound edges while the [Macbeth-Rasputin] match has more partial concepts with high completeness percentages.

The secondary alignment process results in 18 new binding edges between the 7 characters, and 5 secondary alignment groups, which together produce 12 interim outcomes. After validating each and applying commonsense, OPERA produces the following outcome prediction:

```
"Macduff harms Macbeth, who becomes incapacitated."
```

We can see the result of this prediction in the elaboration graph in figure 5.1.1b. The parts circled in red show how the results differ from the original elaboration graph in figure 5.1.1a. Specifically, we can see that:

- 2 new plot units are added – “Macduff harms Macbeth” and “Macbeth becomes incapacitated”
- New connections are drawn
    - Between “Macduff harms Macbeth” and “Macbeth becomes incapacitated”
    - Between existing plot units that were previously partial concepts
- 4 new concepts are discovered
    - **Revenge**: Macduff gets revenge on Macbeth
    - **Phyrric Victory**: Macbeth’s victory in becoming king leads to him becoming harmed and incapacitated
    - **Avenge Family:** Macduff avenges Lady Macduff’s death
    - **Mistake because harmed:** Macbeth harming Lady Macduff was a mistake because it leads to Macduff harming Macbeth.

This prediction is also pretty good in terms of the actual ending: Macduff kills Macbeth. While one of the interim predictions contained the prediction *Macduff kills Macbeth*, OPERA had much higher confidence in the prediction *Macduff harms Macbeth*. In looking at the Macbeth story, there is nothing specific that tells the reader that Macduff will kill Macbeth. Therefore, we believe it makes sense that OPERA would have a higher confidence in the *harm* prediction, and given that *harm* is a superset of *kill,* OPERA still had the actual ending contained within its more general prediction.

Outcome Generation for Macduff character

OPERA gets a similar result by generating an outcome for Macduff (which is the other character involved in Macbeth's ending). For reference, we have included following tables and figures in the next pages:

- Table 5.1.1c – the summarized results of the primary alignment for Macduff.
- Table 5.1.1d – the summarized results of the secondary alignment for Macduff.
- Figure 5.1.1c – the elaboration graph of the shortened *Macbeth story* with the ending that OPERA proposes for Macduff.

The primary alignment process results in 4 potential matches – Anthony, Hamlet, Rasputin, and Laertes (see table 5.1.1c). Note that OPERA is much less confident about the matches and predictions about Macduff. This has to do with the fact that Macduff is a minor character in comparison to Macbeth. Out of 54 sentences in the story, Macduff is mentioned in 5 of them while Macbeth is mentioned in 23 of them. In this case, the result is that Macduff was not involved in any concepts and had no desires, and therefore OPERA uses Macduff's partial concepts to find binding edges. Once these binding edges are found, OPERA adds in the plot alignment and continues as usual to secondary alignment. Macduff's status as a minor character results in significantly lower match scores, exacerbated by the fact that the only partial concept OPERA could find for Macduff was revenge.

The secondary alignment process results in 6 new binding edges between the 4 characters and eventually 2 secondary alignment groups (see table 5.1.1d). Note the higher binding edge scores in the secondary alignment; this means the characters match better to each other than to Macduff. This is an added benefit of the secondary alignment groups: even if individual characters are not a good match to a character, a group of them may be able to generalize better to a character (in this case Macduff). This is beneficial because the characters as a group together highlight the commonality they share with Macduff. OPERA produces 3 interim predictions and after validating each and applying commonsense, OPERA produces the following outcome prediction:

```
"Macduff harms Macbeth."
```

We can see the result of this prediction in the elaboration graph in figure 5.1.1c. The parts circled in red show how the results differ from the original elaboration graph in figure 5.1.1a. Specifically, we can see:

- The new plot unit added – “Macduff harms Macbeth”
- New connections are drawn between existing plot units that were previously partial concepts
- 3 new concepts are discovered
    - **Revenge**: Macduff gets revenge on Macbeth
    - **Avenge Family:** Macduff avenges Lady Macduff’s death
    - **Mistake because harmed:** Macbeth harming Lady Macduff was a mistake because it leads to Macduff harming Macbeth.

Here we see that although the result was not exactly the same as the predicted result for Macbeth (*Macduff harms Macbeth, who becomes incapacitated)*, it is quite similar, which is what was expected. We can also see that the Phyrric Victory concept did not get triggered this time, because the *Macbeth becomes incapacitated* part was missing from OPERA's predictions. This experiment we have shown how OPERA makes predictions both in the presence of a lot of information (the case of Macbeth) and significantly less information (the case of Macduff). We believe that similarity of these outcomes also shows that OPERA is getting a good sense of how the story developed because it arrives at the same outcome from both characters that were involved.

Binding Edge	Bound Edges and Weights	Partial Concepts and Score	Binding Edge Score
Macbeth—Rasputin Plot Score: 7	Mistake because harmed	Avenge Family – 57%	**32.5** 12 (bound edges) + 7 (plot alignment) + 13.5 (partial concepts)
		Avenge Family – 57%	
		Avenge Family – 57%	
		Avenge Family – 57%	
	Mistake because unhappy	Answered Prayer – 57%	
		Answered Prayer – 67%	
		Success – 50%	
	Answered prayer	Success – 50%	
		Success – 33%	
		Revenge – 71%	
	Success	Phyrric Victory – 75%	
		Phyrric Victory – 67%	
		Mistake because Unhappy – 67%	
		Mistake because Harmed – 67%	
Macbeth—Claudius Plot Score: 10	Mistake because harmed	Revenge - 71%	**31** 16.5 (bound edges) + 10 (plot alignment) + ~7.5 (partial concepts)
	Mistake because unhappy	Mistake because unharmed - 67%	
		Mistake because unhappy - 67%	
	Success	Success - 50%	
	Answered Prayer	Success - 50%	
	Regicide	Success - 33%	
	Wants to be king	Answered Prayer - 67%	
Macbeth—Scar Plot Score: 9	Revenge	Phyrric Victory – 67%	**21** 9 (bound edges) + 9 (plot alignment) + 3 (partial concepts)
		Phyrric Victory – 75%	
		Mistake because Harmed – 67%	
	Answered Prayer	Mistake because unhappy – 67%	
		Success – 50%	
		Success – 50%	
	Success	Success – 50%	
		Success – 33%	
		Answered Prayer – 67%	
Macbeth—Hamlet Plot Score: 9	Success	Revenge - 71%	**21.5** 9 (bound edges) + 9 (plot alignment) + ~3.5 (partial concepts)
		Revenge - 71%	
	Answered Prayer	Revenge - 71%	
		Revenge - 71%	
	Regicide	Success - 50%	
		Success - 50%	
Macbeth—Simba Plot Score: 7	Answered Prayer	Success – 50%	**16.5** 6 (bound edges) + 7 (plot alignment) + 3.5 (partial concepts)
		Success – 50%	
		Success – 50%	
	Success	Success – 50%	
		Success – 33%	
		Answered Prayer – 67%	

Macbeth—Cassius Plot Score: 4	Mistake because harmed	Revenge - 71%	**18** 6 (bound edges) + 4 (plot alignment) + ~8 (partial concepts)
		Mistake because harmed - 67%	
	Mistake because unhappy	Mistake because unhappy - 67%	
		Answered Prayer - 67%	
Macbeth—Caesar Plot Score: 1	Answered Prayer	--	**4** 3 (bound edges) + 1 (plot alignment) + 0 (partial concepts)

Table 5.1.1a: Summary of primary alignment results for outcome prediction on Macbeth character.

<table>
<tr><th colspan="2">Binding Edges and Score</th><th>Secondary Alignment Groups</th><th>Grouping Score</th><th>Match Score</th></tr>
<tr><td>Claudius–Scar</td><td>33.5</td><td rowspan="5">Cassius Claudius Hamlet</td><td rowspan="5">16.33
= 1/3*[
17 (Cassius-Claudius)
+14 (Cassius-Hamlet)
+15 (Hamlet-Claudius)]</td><td rowspan="5">23.5
1/3[18 (Cassius-Macbeth)
+31 (Claudius-Macbeth)
+ 21.5(Hamlet-Macbeth)]</td></tr>
<tr><td>Scar-Rasputin</td><td>22</td></tr>
<tr><td>Claudius–Rasputin</td><td>20</td></tr>
<tr><td>Cassius-Scar</td><td>18</td></tr>
<tr><td>Claudius–Hamlet</td><td>15</td></tr>
<tr><td>Hamlet-Simba</td><td>18</td><td rowspan="5">Cassius Claudius Rasputin</td><td rowspan="5">15.33
= 1/3*[
17 (Cassius-Claudius)
+15 (Cassius-Rasputin)
+14 (Hamlet-Rasputin)]</td><td rowspan="5">27.17
1/3[18 (Cassius-Macbeth)
+31 (Claudius-Macbeth)
+ 32.5(Rasputin-Macbeth)]</td></tr>
<tr><td>Cassius–Claudius</td><td>17</td></tr>
<tr><td>Hamlet-Scar</td><td>17</td></tr>
<tr><td>Cassius-Rasputin</td><td>15</td></tr>
<tr><td>Cassius–Hamlet</td><td>14</td></tr>
<tr><td>Hamlet-Rasputin</td><td>14</td><td rowspan="3">Claudius Scar</td><td rowspan="3">16.33
(Claudius-Scar)</td><td rowspan="3">26
1/2[21 (Scar-Macbeth)
+31 (Claudius-Macbeth)]</td></tr>
<tr><td>Scar-Simba</td><td>14</td></tr>
<tr><td>Claudius-Simba</td><td>11</td></tr>
<tr><td>Simba-Rasputin</td><td>10</td><td rowspan="3">Hamlet Simba</td><td rowspan="3">18
(Hamlet-Simba)</td><td rowspan="3">19
1/2[16.5 (Simba-Macbeth)
+ 21.5(Hamlet-Macbeth)]</td></tr>
<tr><td>Claudius-Caesar</td><td>5</td></tr>
<tr><td>Scar-Caesar</td><td>5</td></tr>
<tr><td>Simba-Caesar</td><td>4</td><td>Caesar</td><td>0</td><td>4.0 (Caesar-Macbeth)</td></tr>
</table>

Table 5.1.1b: Summary of secondary alignment results for outcome prediction on Macbeth character.

Binding Edge	Bound edges and Weights	Partial Concepts and Score	Binding Edge Score
Macduff-Anthony Plot Score: 3	--	Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71%	**5** 0 (bound edges) + 3 (plot alignment) + 2 (partial concepts)
Macduff—Hamlet Plot Score: 4	--	Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71%	**6** 0 (bound edges) + 4 (plot alignment) + 2 (partial concepts)
Macduff-Rasputin Plot Score: 3	--	Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71%	**5** 0 (bound edges) + 3 (plot alignment) + 2 (partial concepts)
Macduff-Laertes Plot Score: 4	--	Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71% Revenge – 71%	**6** 0 (bound edges) + 4 (plot alignment) + 2 (partial concepts)

Table 5.1.1c: Summary of primary alignment results for outcome prediction on Macduff character.

Binding Edges and Score		Secondary Alignment Groups	Grouping Score	Match Score
Rasputin-Hamlet	14	Laertes Anthony Hamlet	**12** 1/3*[9 (Laertes-Anthony) +14 (Laertes-Hamlet) +13 (Hamlet-Anthony)]	**5.67** 1/3*[6 (Laertes-Macduff) +5 (Anthony-Macduff) + 6(Hamlet-Macduff)]
Rasputin-Anthony	7			
Rasputin-Laertes	9			
Laertes-Anthony	9	Rasputin Laertes Hamlet	**12.33** 1/3*[14 (Laertes-Hamlet) +14 (Laertes-Rasputin) +13 (Hamlet-Rasputin)]	**5.67** 1/3*[6 (Laertes-Macduff) +5 (Anthony-Macduff) + 6(Hamlet-Macduff)]
Laertes-Hamlet	14			
Anthony-Hamlet	13			

Table 5.1.1d: Summary of secondary alignment results for outcome prediction on Macduff character.

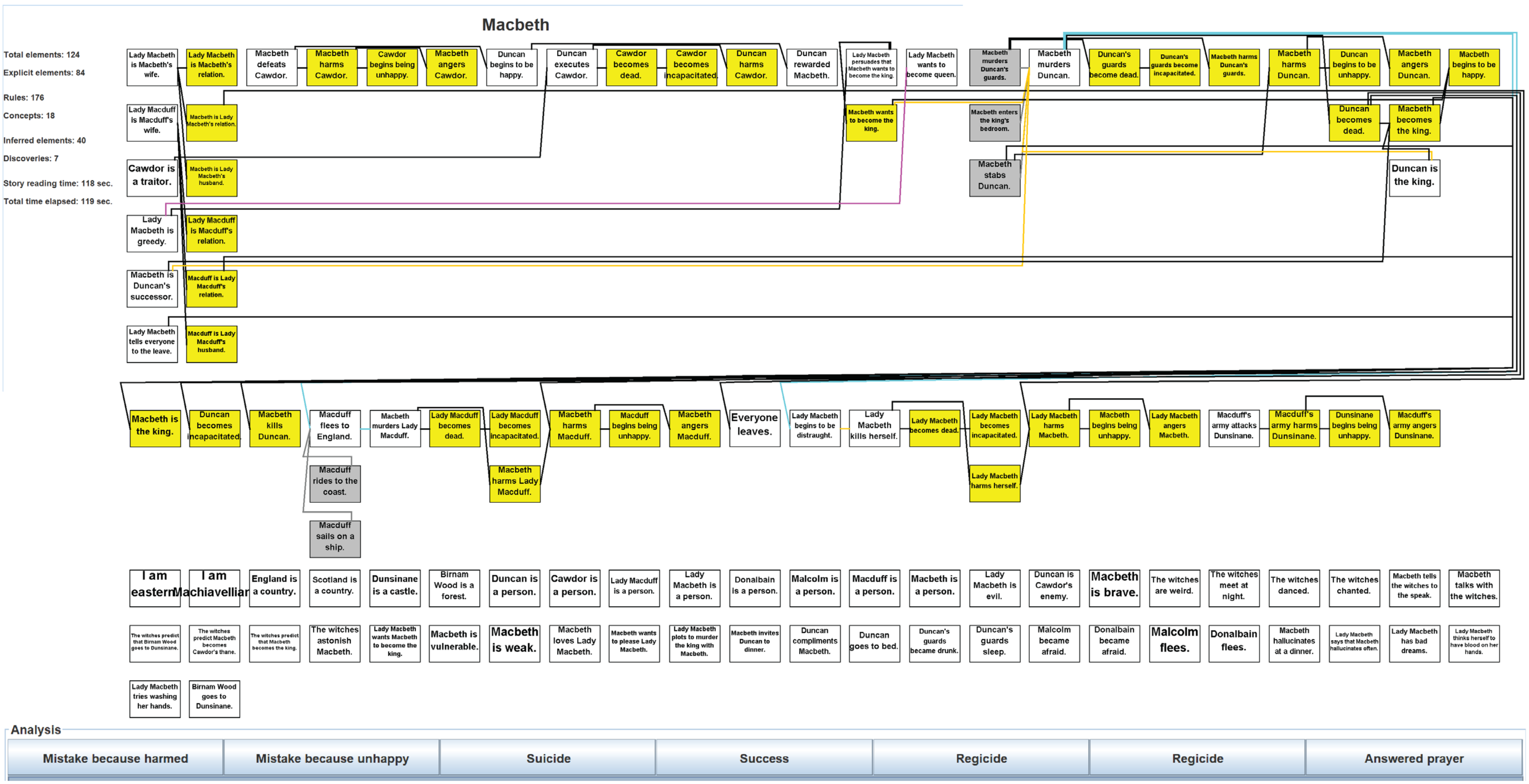


Figure 5.1.1a: Genesis's elaboration graph for original reding of *Macbeth*.

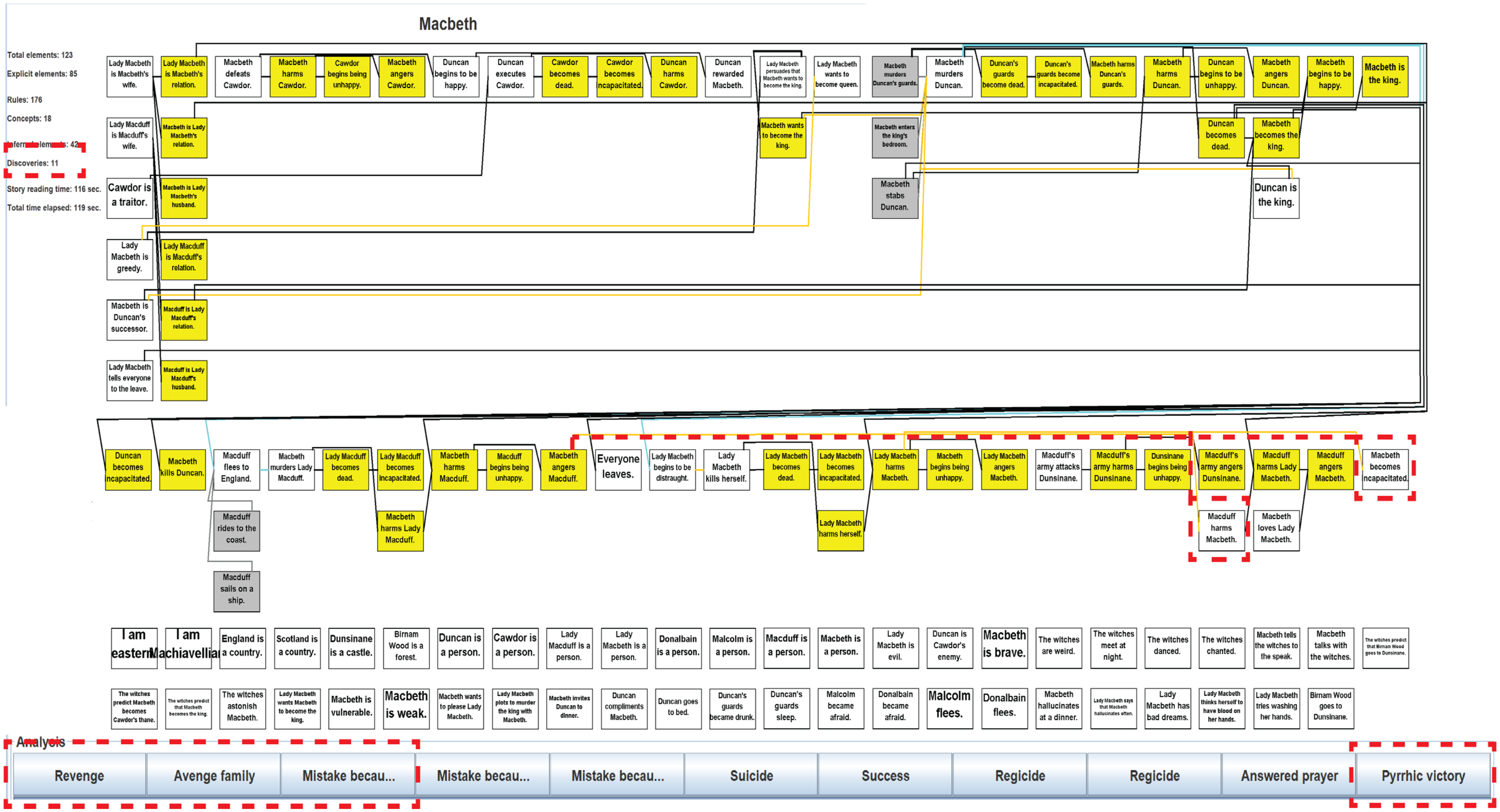


Figure 5.1.1b: Genesis's elaboration graph for *Macbeth* modified with OPERA's prediction for Macbeth: "Macduff harms Macbeth, who becomes incapacitated". Red dashed lines/boxes indicate the changes we see resulting from the modification (when compared to figure 5.1.1a). Specifically note that 4 new concepts are discovered by Genesis including Avenge family, Revenge, Phyrric Victory, and Mistake because Harmed.

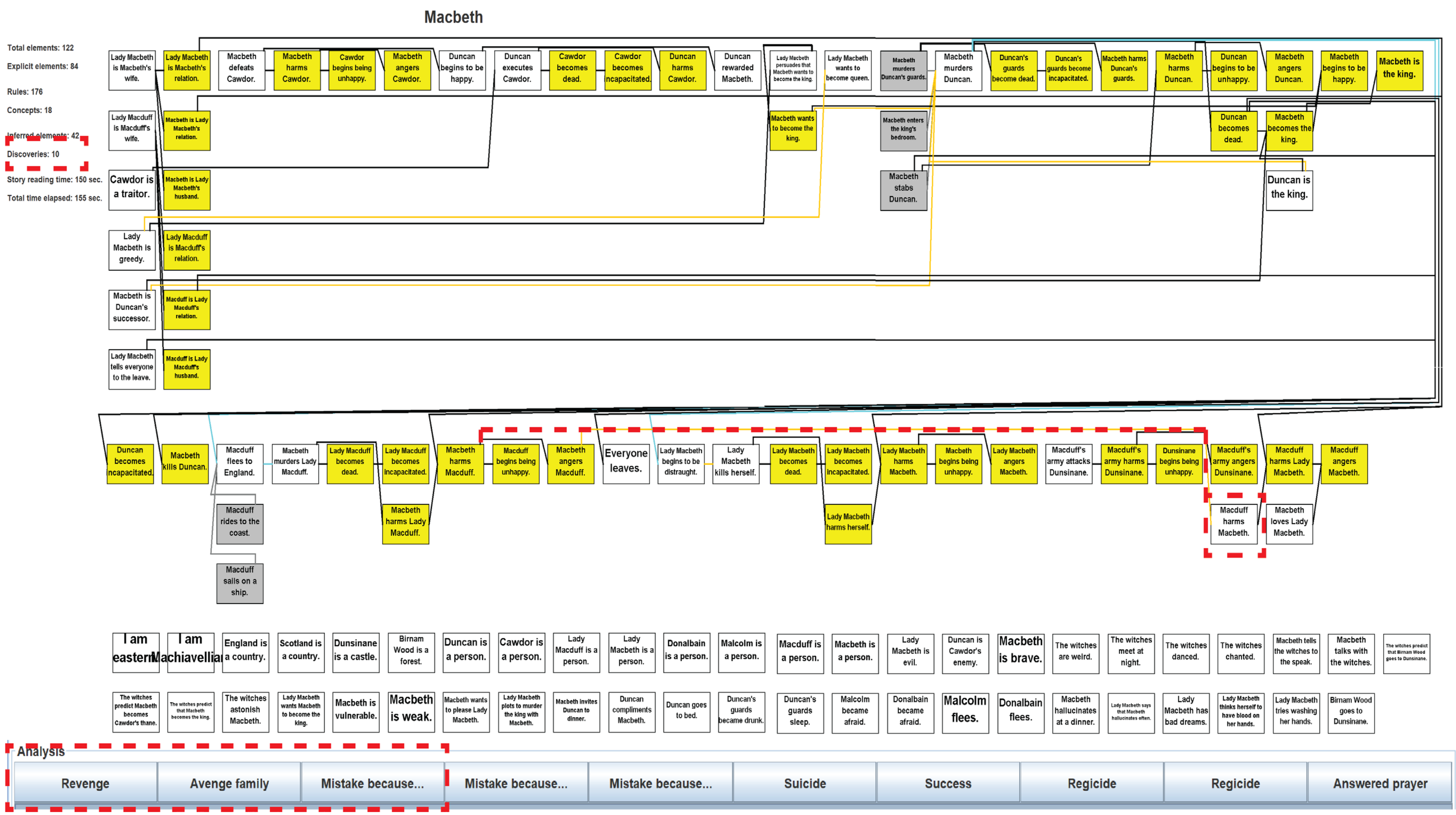


Figure 5.1.1c: Genesis's elaboration graph for *Macbeth* modified with OPERA's prediction for Macduff: "Macduff harms Macbeth". Red dashed lines/boxes indicate the changes we see resulting from the modification (when compared to figure 5.1.1a). Specifically note that 3 new concepts are discovered by Genesis including Avenge family, Revenge, and Mistake because Harmed.

5.1.2 Experiment 2: OPERA's Prediction Changes Based on Where Stories are Ended.

In another experiment, we examine how OPERA's prediction changes based on where the story is stopped. In this experiment, OPERA has read in *Anastasia, Hamlet,* and *Macbeth* as background, and is currently reading *Lion King* with the ending removed (see appendix B4 for *Lion King* with removed ending). Figure 5.1.2a shows the elaboration graph resulting from Genesis reading *Lion King*. The StoryWeb breakdown prior to the outcome generation process is:

Total # of nodes = 50, total # of edges = 52.	
# of DESIRE edges = 14	# of CHARACTER nodes = 29
# of BINDING edges = 0	# of DESIRE nodes = 12
# of CONCEPT edges = 38	# of CONCEPT nodes = 9

Outcome Generation for Scar character

For reference, we have included following tables and figures in the next pages:

- Table 5.1.2a – the summarized results of the primary alignment for Scar.
- Table 5.1.2b – the summarized results of the secondary alignment for Scar.
- Figure 5.1.2b – the elaboration graph of the shortened *Lion King* story with the ending that OPERA proposes for Scar.

The primary alignment process results in 5 potential matches – Rasputin, Claudius, Macbeth, Hamlet, and Macduff (see table 5.1.2a). Here we see the same phenomena as in experiment 1: OPERA most closely matched Scar, who is considered "bad" character, with other "bad" characters – Rasputin, Claudius, Macbeth, and Hamlet. You can also see there is a substantial number of partial concepts matched to each binding edge. This is caused by the fact that we cut off Lion King right before the climax, with a good amount of the story remaining afterwards (9 sentences). The means much of the resolution of the story (and most likely the completion of multiple concepts) was removed as well.

The secondary alignment process results in 9 new binding edges between the 5 characters and eventually 4 secondary alignment groups, which together produce 5 interim outcomes. After validating each and applying commonsense, OPERA produces the following outcome prediction:

```
"Simba harms Scar, who becomes incapacitated."
```

We can see the result of this prediction in the elaboration graph in figure 5.1.2b. Specifically, we can see:

- The new plot unit added – “Simba harms Scar” and “Scar becomes incapacitated”
- New connections are drawn
    - Between “Simba harms Scar” and “Scar becomes incapacitated”
    - Between existing plot units that were previously partial concepts
- 3 new concepts are discovered
    - **Revenge**: Macduff gets revenge on Macbeth
    - **Phyrric Victory:** Scar wanting to become king leads to Scar becoming king, which makes him happy but eventually leads to him becoming incapacitated.
    - **Phyrric Victory:** Scar wanting to get rid of Simba leads to Simba running away which makes Scar happy but eventually leads to Simba harming Scar.
    - **Mistake because harmed:** Scar wanting to get rid of Simba was a mistake because it leads to Simba harming Scar.

This prediction also highlights some of the areas where OPERA can be improved. This prediction is good in the sense that we know that in the next part of the story, Simba and Scar fight and Simba does indeed harm Scar. However, OPERA seems to be implying that Scar becomes incapacitated because Simba harms him, when in the actual story that is caused by the hyenas harming (and killing) Scar. OPERA had an interim prediction "the hyenas harm scar, who becomes incapacitated" but the chosen prediction scored much higher and overruled this one. This is one area where we see the tradeoff between a surprise and expected ending. Everything in the story points to Simba fighting Scar and Simba getting revenge by killing Scar, but in the end, Simba spares Scar (although Scar is later killed by the hyenas). So, the question becomes, what should we expect OPERA to do in this situation? Should it go with the expected outcome? Is there something in the nuances of Simba's personality that would lead it to think Simba would not kill Scar? We discuss this in future work, see below.

Outcome Generation for Simba character

For reference, we have included following tables and figures in the next pages:

- Table 5.1.2c – the summarized results of the primary alignment for Simba.
- Table 5.1.2d – the summarized results of the secondary alignment for Simba.
- Figure 5.1.2b – the elaboration graph of the shortened *Lion King* story with the ending that OPERA proposes for Simba.

The primary alignment process results in 6 potential matches – Rasputin, Claudius, Macbeth, Hamlet, Macduff, and Laertes (see table 5.1.2c). As with Macduff in experiment 1, we can see that Simba has not actually participated in any concepts or desires. Simba is not by any means a minor character, but rather most of Simba's concepts get resolved after the climax. Therefore, given that we cut the story before the climax, OPERA needs to use the partial concepts to find the matches. As with Scar, we see that there are a substantial number of partial concepts that are matched to each binding edge.

The secondary alignment process results in 11 new binding edges between the 6 characters and eventually 5 secondary alignment groups, which together produce 4 interim outcomes. After validating each and applying commonsense, OPERA produces the following outcome prediction:

```
"Simba becomes the king."
```

We did not include the resulting elaboration graph because there is no change in the number of discoveries or connections (only the new outcome is added). This prediction is particularly interesting because it shows how OPERA can predict the outcome of a character beyond the point where the story was cut off. Given that the story was stopped right before the climax, we believed OPERA would predict something about the fight or Simba/Scar harming each other. While OPERA did have the prediction that "Simba harms Scar" in its interim predictions, the prediction "Simba becomes the king" was weighted much higher. We believe this shows that OPERA was able to identify broader themes and ideas in *The Lion King* when compared to *Anastasia, Hamlet,* and *Macbeth* and generate a prediction about Simba's overall outcome in the story beyond the fight with Scar.

Shifting the Story Forward and Analyzing Outcome Generation for Scar and Simba.

We analyze how much the climax of the story impacts our predictions. We did this by preparing a new version of the story that ended a little further along, this time keeping in several sentences containing the climax of the story. We were a bit surprised that the resulting outcome prediction barely changed for both characters.

For Scar, the primary alignment remained mostly the same, except with a couple fewer partial concepts. OPERA did change the final prediction slightly, only predicting that "Scar becomes incapacitated". This is slightly better in terms that it does not link Scar becoming incapacitated to Scar being harmed by Simba, but still does not manage to capture the actual cause (the hyenas). For Simba, the primary alignment resulted in stronger matches because now Simba had a full concept, Mistake because Harmed, on which it could match to other characters with. However, this ended up not impacting the final prediction, which remained the same (Simba becomes king).

This result shows an interesting issue with how OPERA processes a story. On the one hand, it shows how OPERA takes a more general view of the story because it is not greatly affected by a bit of new information. With the climax added in, OPERA adjusted its matches but proceeded mostly as before. On the other hand, we would have expected that even if it were only a little bit of information, given that it was the climax of the story, it would have had a bigger impact on OPERA given that the climax starts resolving some of the storylines for the characters. We noted that the number of edges created remained the same and that only a few more partial concepts were identified. We believe some of our ideas for future work (see Looking Forward section) such as a dynamic weighting scheme and robust commonsense model may be able to help with this issue. These additional features could help OPERA to extract more nuanced, story specific information, enabling it to make better predictions.

Binding Edge	Bound edges and Weights	Partial Concepts and Score	Binding Edge Score
Scar-Claudius Plot Score: 10	Regicide	Answered Prayer – 67%	**36** 16.5 (bound edges) + 10 (plot alignment) + 9.6 (partial concepts)
		Answered Prayer – 67%	
		Answered Prayer – 67%	
	Answered Prayer	Answered Prayer – 67%	
		Mistake because harmed – 67%	
		Mistake because harmed – 67%	
	Success	Mistake because harmed – 67%	
		Mistake because unhappy – 67%	
		Mistake because unhappy – 67%	
	Mistake because harmed	Mistake because unhappy – 67%	
		Revenge – 71%	
		Success – 33%	
	Mistake because unhappy	Success – 33%	
		Success – 33%	
		Success – 33%	
	To become the king	Success – 50%	
		Success – 50%	
		Success – 50%	
Scar-Macbeth Plot Score: 8	Regicide	Answered Prayer – 67%	**34.3** 15 (bound edges) + 8 (plot alignment) + 11.3 (partial concepts)
		Answered Prayer – 67%	
		Answered Prayer – 67%	
	Answered Prayer	Answered Prayer – 67%	
		Avenge Family – 57%	
		Mistake because harmed – 67%	
		Mistake because harmed – 67%	
	Success	Mistake because harmed – 67%	
		Mistake because unhappy – 67%	
		Mistake because unhappy – 67%	
		Mistake because unhappy – 67%	
	Mistake because harmed	Revenge – 71%	
		Success – 33%	
		Success – 33%	
		Success – 33%	
	Mistake because unhappy	Success – 33%	
		Success – 50%	
		Success – 50%	
		Success – 50%	
Scar-Rasputin Plot Score: 5	Answered Prayer	Answered prayer - 67%	**28.5** 12 (bound edges) + 5 (plot alignment) + 11.5 (partial concepts)
		Answered prayer - 67%	
		Answered prayer - 67%	
		Answered prayer - 67%	
		Answered prayer - 67%	
		Answered prayer - 67%	
		Avenge family - 43%	

	Success	Avenge family - 57%	
		Mistake because harmed - 67%	
		Mistake because harmed - 67%	
		Mistake because harmed - 67%	
		Mistake because harmed - 67%	
		Mistake because unhappy - 67%	
		Mistake because unhappy - 67%	
	Mistake because unhappy	Mistake because unhappy - 67%	
		Mistake because unhappy - 67%	
		Pyrrhic victory - 67%	
		Pyrrhic victory - 67%	
		Pyrrhic victory - 67%	
		Pyrrhic victory - 75%	
		Pyrrhic victory - 75%	
	Mistake because harmed	Success - 33%	
		Success - 33%	
		Success - 33%	
		Success - 33%	
		Success - 50%	
		Success - 50%	
		Success - 50%	
`Scar-Hamlet` Plot Score: 3	Regicide	Answered prayer - 67%	**17.6** 9 (bound edges) + 3 (plot alignment) + 5.6 (partial concepts)
		Answered prayer - 67%	
		Answered prayer - 67%	
	Answered Prayer	Revenge - 71%	
		Success - 33%	
		Success - 33%	
		Success - 33%	
	Success	Success - 50%	
		Success - 50%	
		Success - 50%	
`Scar-Macduff` Plot Score: 2	Regicide	--	**5** 3 (bound edges) + 2 (plot alignment) + 0 (partial concepts)

Table 5.1.2a: Summary of primary alignment results for outcome prediction on Scar character.

Binding Edges and Score		Secondary Alignment Groups	Grouping Score	Match Score
Macduff-Rasputin	7	Macbeth Claudius	**33** (Macbeth-Claudius)	**35.15** 1/2[36 (Claudius-Scar) +34.3 (Macbeth-Scar)
Macduff-Claudius	5			
Macduff-Hamlet	15			
Rasputin-Macbeth	21			
Rasputin-Hamlet	14	Macbeth Hamlet	**23** (Macbeth-Hamlet)	**24.45** 1/2[34.3 (Macbeth-Scar) +14.6 (Hamlet-Scar)]
Rasputin-Claudius	20			
Claudius-Hamlet	18			
Claudius-Macbeth	33	Rasputin	**0**	28.5 (Rasputin-Scar)
Hamlet-Macbeth	23	Macduff	**0**	**5.0** (Macduff-Scar)

Table 5.1.2b: Summary of secondary alignment results for outcome prediction on Scar character.

Binding Edge	Bound edges and Weights	Partial Concepts and Score	Binding Edge Score
`Simba-Rasputin` Plot Score: 2	--	Answered prayer - 67% Answered prayer - 67% Answered prayer - 67% Answered prayer - 67% Mistake because harmed - 67% Mistake because harmed - 67% Mistake because unhappy - 67% Mistake because unhappy - 67% Mistake because unhappy - 67% Pyrrhic victory - 58% Revenge - 71% Revenge - 71% Revenge - 71% Revenge - 71% Success - 33% Success - 33% Success - 33% Success - 33%	**13** 0 (bound edges) + 2 (plot alignment) + 11 (partial concepts)
`Simba-Hamlet` Plot Score: 6	--	Answered prayer - 67% Answered prayer - 67% Answered prayer - 67% Answered prayer - 67% Revenge - 71% Revenge - 71% Revenge - 71% Revenge - 71% Success - 33% Success - 33% Success - 33% Success - 33%	**11** 0 (bound edges) + 6 (plot alignment) + 5 (partial concepts)
`Simba-Macduff` Plot Score: 4	--	Revenge - 71% Revenge - 71% Revenge - 71% Revenge - 71%	**6.5** 0 (bound edges) + 4 (plot alignment) + 2.5 (partial concepts)
`Simba-Macbeth` Plot Score: 4	--	Answered prayer - 67% Answered prayer - 67% Answered prayer - 67% Answered prayer - 67% Mistake because harmed - 67% Mistake because harmed - 67% Mistake because harmed - 67% Mistake because unhappy - 67% Mistake because unhappy - 67%	**10** 0 (bound edges) + 4 (plot alignment) + 6 (partial concepts)

		Mistake because unhappy - 67%	
		Success - 33%	
		Success - 33%	
		Success - 33%	
		Success - 33%	
Simba-Claudius Plot Score: 3	--	Answered prayer - 67%	**9** 0 (bound edges) + 2 (plot alignment) + 7 (partial concepts)
		Answered prayer - 67%	
		Answered prayer - 67%	
		Answered prayer - 67%	
		Mistake because harmed - 67%	
		Mistake because harmed - 67%	
		Mistake because harmed - 67%	
		Mistake because unhappy - 67%	
		Mistake because unhappy - 67%	
		Mistake because unhappy - 67%	
		Success - 33%	
		Success - 33%	
		Success - 33%	
		Success - 33%	
Simba-Laertes Plot Score: 5	--	Revenge - 71%	**7** 0 (bound edges) + 5 (plot alignment) + 2 (partial concepts)
		Revenge - 71%	
		Revenge - 71%	
		Revenge - 71%	

Table 5.1.2c: Summary of primary alignment results for outcome prediction on Simba character.

Binding Edges and Score		**Secondary Alignment Groups**	**Grouping Score**	**Match Score**
Macduff-Rasputin	7	Macbeth Claudius	**33** (Macbeth-Claudius)	**9.5** 1/2[10 (Macbeth-Simba) +9 (Claudius-Simba)]
Macduff-Claudius	5			
Macduff-Hamlet	15			
Macduff-Laertes	8	Macbeth Hamlet	**23** (Macbeth-Hamlet)	**10.5** 1/2[10 (Macbeth-Simba) +11 (Hamlet-Simba)]
Rasputin-Macbeth	21			
Rasputin-Hamlet	14	Rasputin	**0**	**13** (Rasputin-Simba)
Rasputin-Laertes	9			
Rasputin-Claudius	20	Laertes	**0**	**7** (Laertes-Simba)
Claudius-Hamlet	18			
Claudius-Macbeth	33	Macduff	**0**	**6.5** (Macduff-Simba)
Hamlet-Macbeth	23			

Table 5.1.2d: Summary of secondary alignment results for outcome prediction on Simba character.

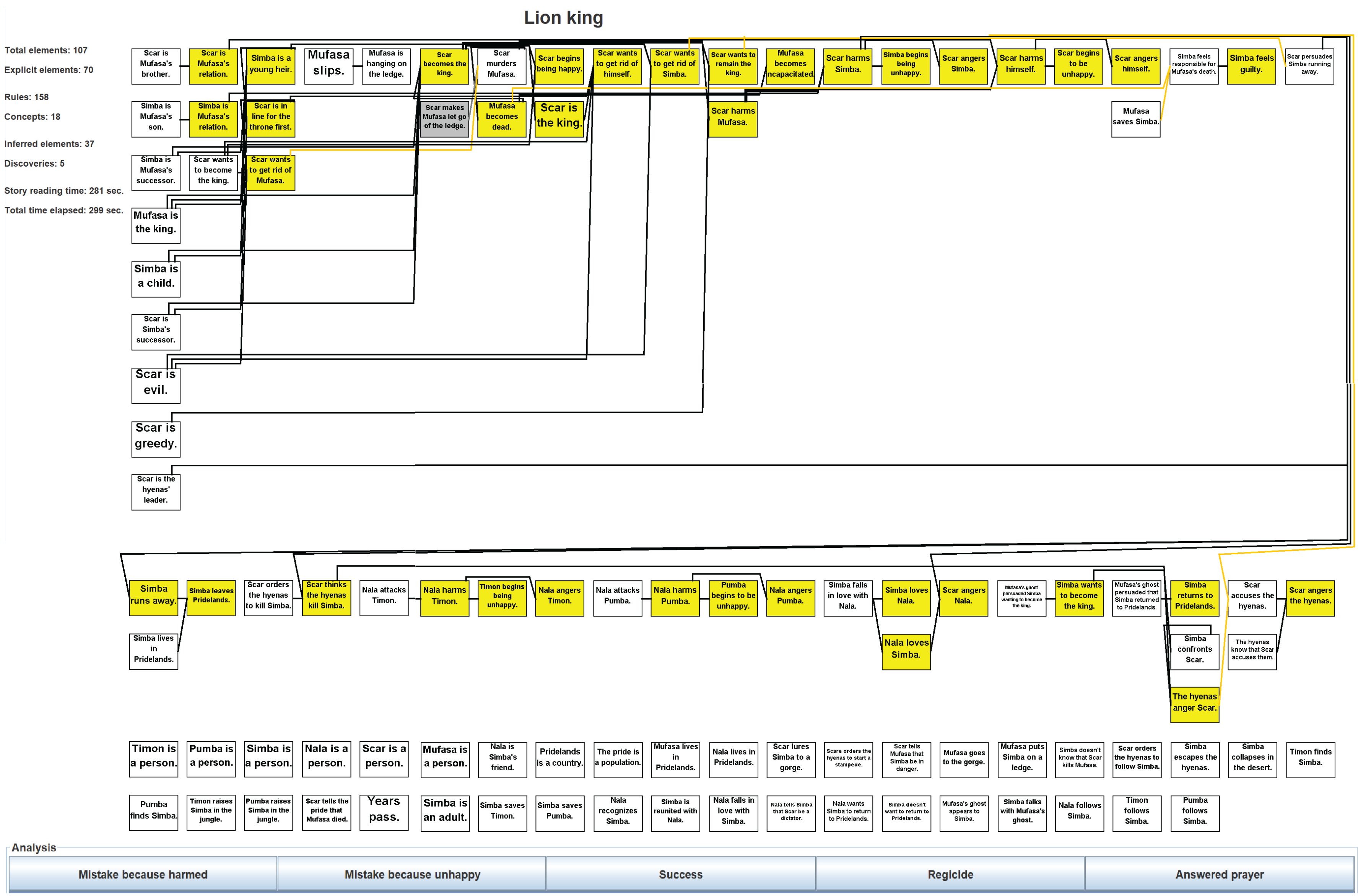


Figure 5.1.2a: Genesis's elaboration graph for *Lion King*.

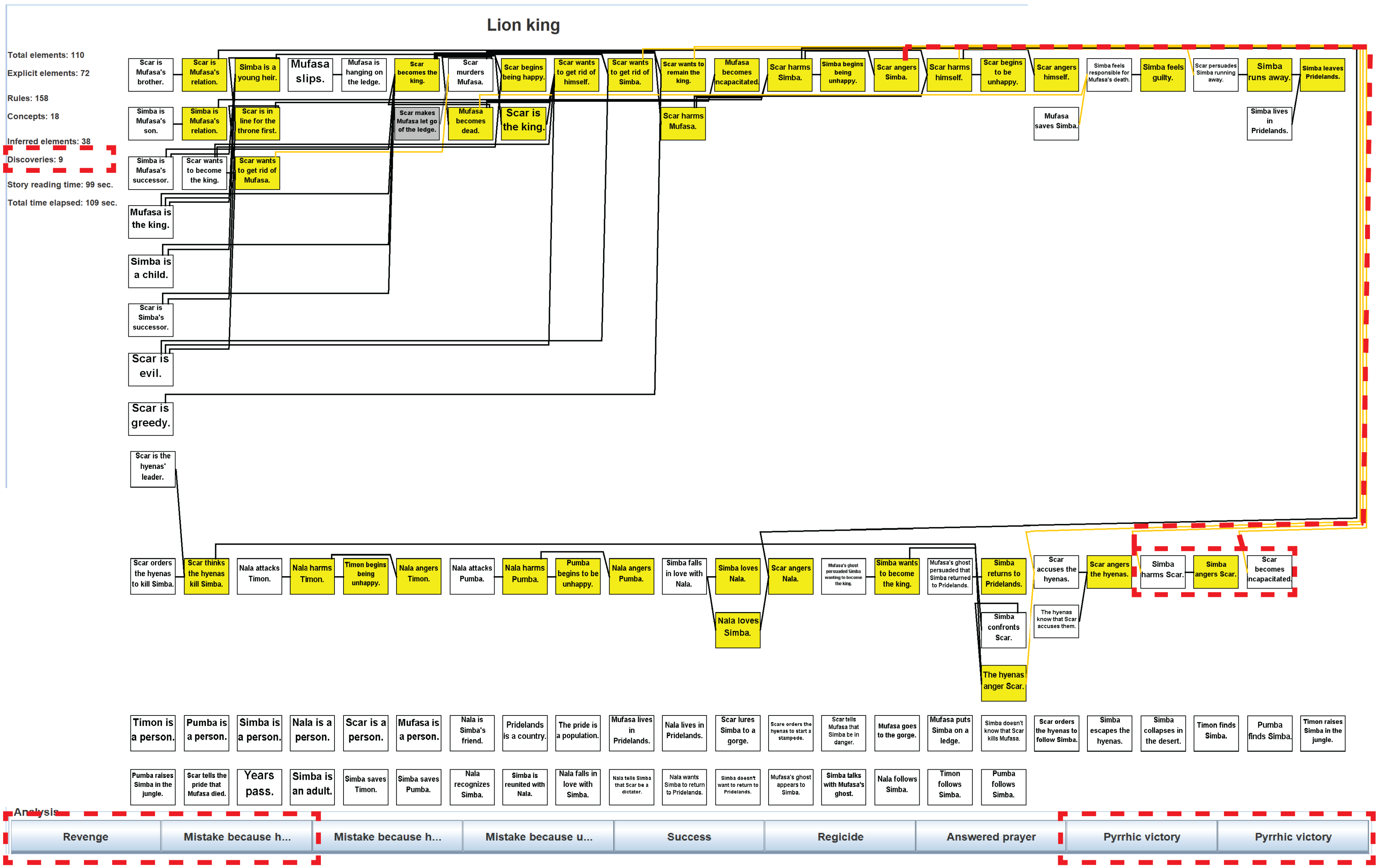


Figure 5.1.2b: Genesis's elaboration graph for *Lion King* modified with OPERA's prediction for Scar: "Simba harms Scar, who becomes incapacitated". Red dashed lines/boxes indicate the changes we see resulting from the modification (when compared to figure 5.1.2a). Specifically note that 4 new concepts are discovered by Genesis including Revenge, Phyrric Victory, and Mistake because Harmed.

5.1.3 Chapter 4 Example

This section presents in detail the example used in chapter 4. In this example, OPERA read *Hamlet* and *Julius Caesar* as background, and is currently reading *Macbeth* with the ending removed (see appendix B5 for *Macbeth* with removed ending). Figure 5.1.1a shows the elaboration graph resulting from Genesis reading *Macbeth*. The StoryWeb breakdown prior to the outcome generation process is:

Total # of nodes = 33, total # of edges = 34.	
# of DESIRE edges = 6	# of CHARACTER nodes = 19
# of BINDING edges = 0	# of DESIRE nodes = 6
# of CONCEPT edges = 28	# of CONCEPT nodes = 8

For reference, we have repeated the following tables in next pages:

- Table 4.3.4b – the full results of the primary alignment for Macbeth
- Table 4.3.5a and 4.3.5c – the full results of the secondary alignment for Macbeth

The primary alignment process results in 4 potential matches – Claudius, Hamlet, Cassius, and Caesar (see table 4.3.4b). The secondary alignment process results in 3 new binding edges between the 4 characters and eventually 2 secondary alignment groups (see table 4.3.5a and 4.3.5c). These 2 groups and the 4 original binding edges produce 6 interim outcomes. After validating each and applying commonsense, OPERA produces the following outcome prediction:

```
"Macduff harms Macbeth."
```

We can see the result of this prediction in the elaboration graph in figure 5.1.1c. The parts circled in red show how the results differ from the original elaboration graph in figure 5.1.1a. Specifically, we can see the new plot units added, the new connections that are drawn, and most importantly, the 3 new concepts that resulted from the addition (Revenge, Avenge Family, and Mistake because Harmed).

One interesting thing to note here is that despite the reduced number of stories read in comparison to experiment 1 in 5.1.1, we still get part of the predicted outcome, *Macduff harms Macbeth,* from that experiment. This shows us that the other two stories read in experiment 1 – *Lion King* and *Anastasia* – were the ones that contributed the part of *Macbeth becomes incapacitated.* Therefore, even if the stories match well to each other, it is still beneficial to read more stories as they may contribute more to the prediction.

Binding Edge	Bound edges and Weights	Partial Concepts and Score		Binding Edge Score
Macbeth—Cassius Plot Score: 4	Mistake because harmed (CE) – 3	Revenge - 71%	2.13 = 3*0.71	**18** 6 (bound edges) + 4 (plot alignment) + ~8 (partial concepts)
		Mistake because harmed - 67%	2.01 = 3*0.67	
	Mistake because unhappy (CE) – 3	Mistake because unhappy - 67%	2.01 = 3*0.67	
		Answered Prayer - 67%	2.01 = 3*0.67	
Macbeth—Claudius Plot Score: 10	Mistake because harmed (CE) – 3	Revenge - 71%	2.13 = 3*0.71	**36** 16.5 (bound edges) + 10 (plot alignment) + ~9.5 (partial concepts)
	Mistake because unhappy (CE) – 3	Mistake because unharmed - 67%	2.01 = 3*0.67	
		Mistake because unhappy - 67%	2.01 = 3*0.67	
	Success (CE) – 3	Success - 50%	1.50 = 3*0.50	
	Answered Prayer (CE) – 3	Success - 50%	--	
	Regicide (CE) – 3	Success - 33%	--	
	Wants to be king (DE) – 1.5	Answered Prayer - 67%	2.01 = 3*0.67	
Macbeth—Caesar Plot Score: 1	Answered Prayer (CE) – 3	--	--	**4** 3 (bound edges) + 1 (plot alignment) + 0 (partial concepts)
Macbeth—Hamlet Plot Score: 9	Success (CE) – 3	Revenge - 71%	2.13 = 3*0.71	**21.5** 9 (bound edges) + 9 (plot alignment) + ~3.5 (partial concepts)
		Revenge - 71%	--	
	Answered Prayer (CE) – 3	Revenge - 71%	--	
		Revenge - 71%	--	
	Regicide (CE) – 3	Success - 50%	1.50 = 3*0.50	
		Success - 50%	--	

Table 4.3.4b: Binding edges resulting from OPERA running primary alignment on Shakespeare example part 4: weighting results. OPERA calculates a score for each binding edge by performing a weighted sum across the bound edges, aligned plot, and partial concepts. Note that edges with multiple instances of the same partial concept, such as revenge in the `Macbeth—Hamlet` edge, only count the most complete partial concept in the score (ignored partial concepts have a -- in their cell). (CE=Concept Edge; DE=Desire Edge).

Secondary Binding Edge	Bound edges and Weights	Binding Edge Score
Cassius—Claudius Plot Score: 8	Mistake because harmed (CE) – 3 Mistake because unhappy (CE) – 3 Revenge (CE) – 3	**17** 9 (bound edges) + 8 (plot alignment)
Cassius—Hamlet Plot Score: 8	Answered Prayer (CE) – 3 Revenge (CE) – 3	**14** 6 (bound edges) + 8 (plot alignment)
Claudius—Caesar Plot Score: 2	Answered Prayer (CE) – 3	**5** 3 (bound edges) + 2 (plot alignment)
Claudius—Hamlet Plot Score: 9	Success (CE) – 3 Answered prayer (CE) – 3 Revenge (CE) – 3	**18** 9 (bound edges) + 9 (plot alignment)

Table 4.3.5a: Binding edges resulting from OPERA performing secondary alignment. The four characters that matched to Macbeth in the primary alignment (Cassius, Claudius, Caesar, and Hamlet) are aligned to each other to quantify their similarity. Note that not all characters may match to each other in secondary alignment (such as Caesar who only matched to Claudius). (CE=Concept Edge; DE=Desire Edge).

Validated Binding Edges	Secondary Alignment Groups	Grouping Score	Match Score
Cassius—Claudius Cassius—Hamlet Hamlet—Cassius Hamlet—Claudius	Cassius Claudius Hamlet	**16.33** = 1/3*[17 (Cassius-Claudius) +14 (Cassius-Hamlet) +18 (Hamlet-Claudius)]	**25.16** = 1/3[18 (Cassius-Macbeth) +36 (Claudius-Macbeth) + 21.5(Hamlet-Macbeth)]
Claudius—Cassius Claudius—Hamlet	Caesar	**0**	**4.0** (Caesar-Macbeth)

Table 4.3.5c: Scoring process for secondary alignment groups to evaluated matches. The grouping score quantifies how well the characters in the group match to each other and is calculated from the binding edges created during secondary alignment. The match score quantifies how well the group matches to the main

5.2 Improvements

Throughout this thesis, we have discussed implementation decisions that improve the running of the system. We present them here more formally and introduce additional improvements as well.

The main improvement that underlies much of this work is the decision to focus on character alignment. As discussed in chapter 3, we believe character alignment is a better approach versus story alignment because stories contain a lot of superfluous information not needed to make predictions. As noted earlier [4], the process of aligning stories together while maintaining character continuity is computationally expensive. Throughout chapter 4, we showed how focusing on characters instead of the entire story in OPERA reduced matching problem significantly.

In terms of storage, we presented the graphical StoryWeb data structure in section 4.2 as a vehicle to compactly represent the information OPERA needs to remember and work from. The StoryWeb is kept compact because of the design: the nodes represent abstract ideas while the edges are specific instances. We gave the example of Macbeth, where OPERA reduced 160 story elements to only 18 nodes and 10 edges. In figure 4.2.1, we showed how the number of nodes in the web grows at a significantly slower rate than the edges as more stories are read. In figure 4.2.2, we showed how structure also enabled efficient searching throughout the web, allowing OPERA to find related character nodes via only a pair of steps through the web.

Within the character alignment process, we worked to decrease the cost of computing the plot alignments between two characters. The most computationally expensive part of plot alignment has to do with the fact that there must be character continuity through the alignment, i.e., the same characters should remain matched to each other. To ensure this continuity, all the characters in the plots must be bound to each other before performing the actual alignment. As Fay showed in this work, this becomes increasingly expensive as the size of the stories increases [4]. We indirectly addressed this through the idea of character alignment because character plots are significantly shorter and narrower in scope than the story. More directly, we developed a set of core bindings from the concept and desire matches which we passed into Fay's plot aligner. In doing so, we reduced the number of unmatched bindings that the plot aligner had to search for significantly. For characters that are very similar, the reduction is greater because most of the bindings that are needed for the plot alignment will have already been in the set of core bindings.

The final improvement has to do with the decision to build OPERA as a module on top of the Genesis system. Genesis has a "box and wire" paradigm in which classes can be connected to each other (and therefore communicate) through a system of wires. Most of the sub-systems built in OPERA are implemented with this box and wire paradigm so that they can (1) signal other sub-systems to start working on a specific task, (2) send data to different sub-systems, (3) send results to a sub-system after computation is complete. The Partial Concept Expert is triggered to run on the story while another sub-system is processing the different characters. Once the Partial Concept Expert finishes finding all the partial concept instances, it sends the resulting instances back to the original subsystem to be integrated with the binding edges, when they are created. This allows OPERA to complete tasks asynchronously, taking advantage of the fact that much of the processing for character alignment happens modularly, and is not combined until the end for the final prediction.

6 Contributions

6.1 Contributions

In this thesis, we presented a novel method for character alignment – a symbolic, analogical-reasoning-based approach to prediction. We argued that story understanding naturally lends itself to the task of prediction because it is a symbolic methodology that allows for reasoning about an inner model of the world. We claimed this reasoning should specifically be reasoning by analogy, which enables us to compare past experiences, generalize them, and apply them to current situations.

In order to demonstrate character alignment in action, we designed and developed OPERA – a computational system that expands the Genesis Story Understanding system by making predictions about story outcomes. OPERA creates action-oriented representations for each character it reads about in a story. When making a prediction, OPERA uses a multi-level alignment scheme to compare these representations to each other and quantify their similarity. This scheme enables both direct and secondary comparison, allowing OPERA to group similar characters together and extract common themes. OPERA builds up the prediction in stages from the groups of similar characters and from identified partial concepts. In the final stage, OPERA uses commonsense reasoning from the Genesis system to group predictions that are causally connected to each other before proposing an outcome. With this new ability to reason about and generalize past experiences to make a prediction, OPERA takes the Genesis Story Understanding System to the next level. In the process of achieving all this, we made the following contributions:

We presented character alignment as a novel approach. Character alignment shifts the focus of analysis from the entire story to the individual characters themselves. The key idea is to create representations of characters in each story built up from the concepts character is involved in, their desires and their plot. We showed how focusing on characters is valuable because it reduces the number of elements OPERA needs to look at and eventually compare while still allowing for good predictions.

We described how partial concepts elevate character alignment. Partial concepts give us a sense of what a character might do, and our confidence in a partial concept can be quantified by how instantiated the partial concept is in a story. By augmenting the character alignments with partial concept information, OPERA measured how likely a character was to take an action based on how strongly that character aligned to another character who completed the concept. This functionality was used in generating potential predictions.

We designed a compact data structure that enables quick searching. The graphical StoryWeb data structure is at the base of the OPERA implementation. It was designed such that the different node types represent abstract ideas, and the edges connect specific instances of those ideas to each other. This compact design keeps the size of the StoryWeb small as more stories are read, which allows for the reading of multiple stories.

We developed a multi-level alignment algorithm that enables pattern detection. The primary alignment tells us how past characters relate to the current character, and the secondary alignment tells us how those past characters relate to each other. Together, both alignments enable OPERA to find patterns and group characters together. These groupings provide an abstraction from the specific details of single characters to a more general group of similar characters. It allows OPERA to identify repetitive outcomes or situations within the group and to match those outcomes or situations to the character rather than matching characters individually.

We explained why commonsense knowledge is important in prediction. Stories do not always contain specific information about why certain things happened, and that's perfectly fine because humans' common sense is exceptionally good at inferring causations and explanations. This common sense is critical to prediction because it allows us to reason critically about *why* certain things happened previously, giving us a better understanding of whether that situation is likely to unfold again in the present. OPERA uses Genesis's commonsense reasoning model to achieve this to an extent, but as we noted in the previous section, a more robust model could improve the prediction process.

We demonstrated the role of story understanding in understanding prediction and ultimately, ourselves. Prediction is based on human telling themselves stories. These stories allow us to piece together the things we have seen and the things we know and then to reason critically about what may come. This thesis presented a new approach to prediction by leveraging story understanding and focusing on characters. Inspired by previous work in cognitive science, neuroscience, and computer science and AI, we developed a computational model of prediction in story understanding systems, using it to build a system – OPERA – that exhibited intelligent behavior. OPERA gives Genesis the ability to draw connections from past experiences, recognize similarities and patterns, and make informed predictions, moving it closer to achieving human learning and reasoning.

6.2 Looking Forward

The novel approach to prediction via character alignment and the accompanying OPERA implementation described in this thesis is a step toward achieving human-like prediction in story understanding systems. Looking forward, there are three extension that we believe will provide further steps toward our goal:

- **A more robust model of commonsense reasoning,** which would enable understanding of *why* an event or action occurs– *what* caused it, what is the *goal*, what are possible *results*. This feature is needed when combining the interim predictions because OPERA needs a sense of how they may be related to each other. The current implementation uses Genesis's basic commonsense knowledge, but an interesting and promising alternative is the Aspire system built by Williams, Lieberman, and Winston [25] [26] which integrated ConceptNet's [18] large-scale commonsense database into Genesis. We believe further investigation and potential integration of the Aspire system with OPERA could result in improved predictive ability.

- **A dynamic weighting scheme for alignment,** that adjusts based on how important a concept or desire is. This feature is needed because the presence of certain concepts, such as revenge, in a story does not inherently imply that it is, relevant to a given character's goals. Therefore, our system would benefit from having a weighting scheme that adjusts dynamically so that each story's concepts and desires are weighted based on impact. This scheme is dependent on Genesis's ability to detect which parts of a story are the most important to a specific character or goal. There are two approaches we believe could be useful to this task. One possibility is the counterfactual approach where importance is measured by how much a story changes when a plot unit is removed. A second approach could be story summarizations because summarization requires the identification of important information and compression/deletion of non-crucial information (see Winston's story summarizer [27]). Both approaches could identify which plot units are the most important and could give insight into how concepts and desires should be weighted for the story.

- **Defining what the ground truth expected outcome should be.** This feature is especially interesting in the context of surprise: If a story has a surprise ending, what should the expected behavior be for the prediction? Should the system be expected to predict the surprise ending or the "expected" ending?

A final and more general next step toward improving prediction in story understanding involves extending OPERA to play the role of a *student*, such that it explains why it believes certain things, it can answer questions about its process, and use feedback from the user about the final prediction to update its model. The StoryWeb structure that underlies OPERA provides the structure for adding this extension. These improvements would further enable OPERA to take the Genesis Story Understanding System to the next level. By giving Genesis the ability to draw connections from past experiences, recognize similarities and patterns, and make informed predictions, OPERA moving it closer to achieving human learning and reasoning for prediction.

A. The Genesis Enterprise

In this thesis, we presented character alignment as an approach to prediction in story understanding systems and implemented OPERA as a module on top of the Genesis Story Understanding System specifically. To fully understand character alignment and the inner workings of OEPRA, it is necessary to also understanding the ideas and inner workings behind Genesis. This section discusses the Genesis Enterprise, an endeavor undertaken by Professor Patrick Winston and his colleagues based on the view that story understanding is at the core of other aspects of human intelligence, such as problem solving and predictions. The Genesis Story Understanding System is the product of this endeavor and has been developed as a computational model of human story understanding capabilities.

A.1 Stories and Story Understanding are Uniquely Human

What makes humans different from other species? What is unique about human intelligence? It isn't just our language ability: there are gorillas like Koko that learned and communicated with sign language (albeit with a limited vocabulary) [28]. It is not just our problem-solving skills: there are rats that intelligently navigate and memorize mazes to find food. We could go on, but this boils down to a single question: what makes human intelligence different from other species? We believe that humans are uniquely intelligent because we are symbolic. Being symbolic means that we can form "complex, highly nested symbolic descriptions of classes, properties, relations, actions and events" [2].

A.1.1 MERGE Made Us the Only Symbolic Species

This begs the questions of why us? Why are humans symbolic when no other species are? In Why Only Us: Language and Evolution [1], Berwick and Chomsky argue that the ability that separates human intelligence from the intelligence of other life forms is *Merge.*

> **Merge**: an operation that takes two objects – X and Y – and forms a new object that consist of the set of both X and Y together. Provided with conceptual atoms of the lexicon, the operation Merge, iterated without bound, yields an infinity of digital, hierarchically structured expressions [1].

Biologically, they claim this operation is enabled by the closing of an anatomical loop in the human brain. This same loop is nearly complete in the brains of other primates, but not fully. This is what allowed humans to become hierarchical, building new representations from existing ones.

A.1.2 Being Symbolic Enables Inner Stories

Merge provides proof that there is something unique to the human brain that sets us apart from any other species. We believe that Merge enabled us to become a symbolic species. Importantly, our symbolic ability allows us to form even more complex, highly nested, symbolic descriptions called inner-stories.

> **Inner-story:** A collection of complex, highly nested symbolic descriptions of properties, relations, actions, and events, usefully connected with, for example, causal, means-ends, enablement, and time constraints [2].

Using our inner story, we represent the world around us. It allows us to create new ideas, solve problems, and understand stories. Eventually humans evolved to externalize these inner stories and internalize the stories of other humans, creating a feedback loop in our brain. We can see evidence of this in the early cave paintings dating back almost 70,000 years. Even before any evidence of a formal language, these drawings were made to tell stories about the world.

A.1.3 Inner Stories Enable Story Understanding

This ability to tell ourselves an inner story, externalize it to share with others, and internalize the stories we hear is unique to us.

> **The Strong Story Hypothesis:** The mechanisms that enable humans to tell, to understand, and to recombine stories separate our intelligence from that of other primates [2].

So, if we are to develop a computational model of human intelligence, then we need to model human story-understanding ability. We need to model the process by which we form inner stories, either directly from our own experiences or from stories heard from others, and the process by which we externalize those inner stories to share with others. With these ideas in mind, the Genesis Story Understanding System was created.

A.2 The Genesis Story Understanding System Models Human Intelligence

The Genesis Story Understanding System [29] referred to in short as Genesis, is a system that computationally models human story understanding such as aligning different stories, interpreting stories with cultural biases, and drawing analogies to similar stories. Importantly, new abilities that are added to Genesis follow a computational imperative:

The Computational-Imperative Principle: Any model of human intelligence should introduce only computational capabilities that enable observed behaviors without enabling unobserved behaviors [2]

This means that everything in Genesis is meant to model what humans do naturally, and nothing beyond. Figure A-1 provides an overview of the humanly plausible layers upon which Genesis is built [2]. The following sections describe the implantation pieces of Genesis that OPERA was built on top of

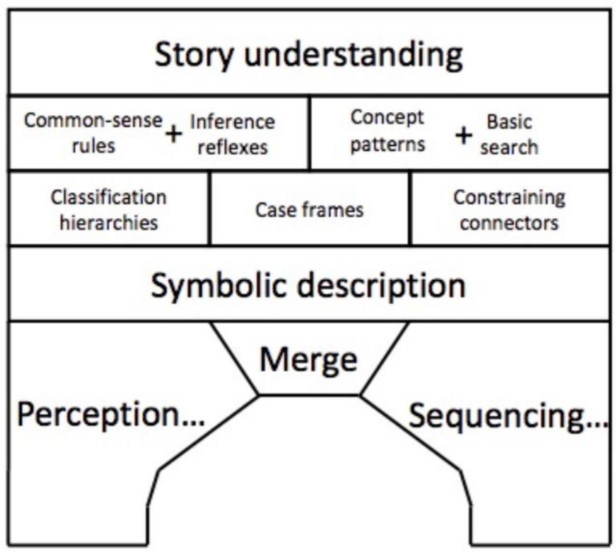


Figure A-1: The Genesis Story Understanding System (Genesis) is a computational model of human story understanding. Genesis is empowered by deductive and inductive rules, providing it with common-sense and inference. Genesis can also recognize concepts by searching stories for patterns using basic search. Going a level deeper, these faculties are empowered by constraints, classification hierarchies, and case frames, which are modeled on our uniquely symbolic capabilities. Figured sourced from [2]

A.2.1 Modeling Inner Stories: English to Innerese

Genesis takes as input a short textual story in simple English, parsing the input using the START parser [30], generating innerese. Innerese is Genesis's inner language whose primary components are Entities, Functions, Relations, and Sequences. These components are modeled after the Java classes such that Entity is a parent class, Function inherits Entity, Relation inherits

Function, and Sequence inherits Entity. We provide a brief description of each representation below; more information can be found in the Genesis implementation substrate [31].

An **entity** is the fundamental building block of innerese. An entity has a unique name and a *bundle of threads*. Threads define the meanings of word and are derived from WordNet [24], a lexical database that captures synonyms, definitions, and the hierarchical relationship of words. A single object is an example of an entity, such as *sandwich.*

A **function** is an entity with the addition of a subject field. A function represents Jackendoff's paths and places elements [32]. Functions usually depict prepositional phrases, such as *next to the sandwich.*

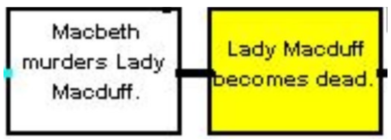
A **relation** depicts how one entity relates to another, such as *Sally steals the sandwich.*

A **sequence** is an ordered set of entities, such as *Sally steals the sandwich from the store.*

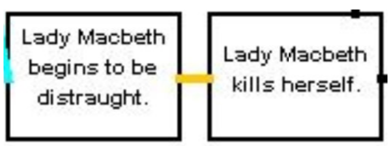
A.2.2 Modeling Common Sense: Rules and Concept Patterns

Stories often imply meanings and consequences rather than directly expressing them. As Genesis reads a story, it uses commonsense knowledge in the form of rules to make inferences that supply missing causal connections, and knowledge in the form of concept patterns to help identify overarching themes. Whenever Genesis applies a commonsense rule, we say that there has been an inference reflex. An inference reflex results in either (1) injection of new elements into the story if not already present and further inference checks, or (2) connection of existing story elements. This is why OPERA sends the story with the new ending back to Genesis to read after the prediction process has been completed; we want to see if Genesis is able to draw any new inferences and expand the story even further given the new element added. Genesis visualizes this process by drawing an elaboration graph (see figure A-2.) We provide a brief example of two rules and their result in the elaboration graph below. More information can be found in The Genesis Enterprise [2].

Deduction Rule: An explicit “If X ... then Y” rule that adds elements to the story if filled. A common example is death, in which we specify rules for when a person becomes dead: “If XX murders YY, then YY becomes dead” results in this connection in the Macbeth story:



Explanation Rule: A “If X…, then Y may…” rule that explains why an event may have happened. Notably explanation rules only draw connections between existing story elements, they **do not add elements** to the story. “If XX becomes distraught, then XX may kill themselves” attempts to explain why XX might have killed themselves, by linking it to XX becomes distraught: “If XX becomes distraught, then XX may kill themselves”.



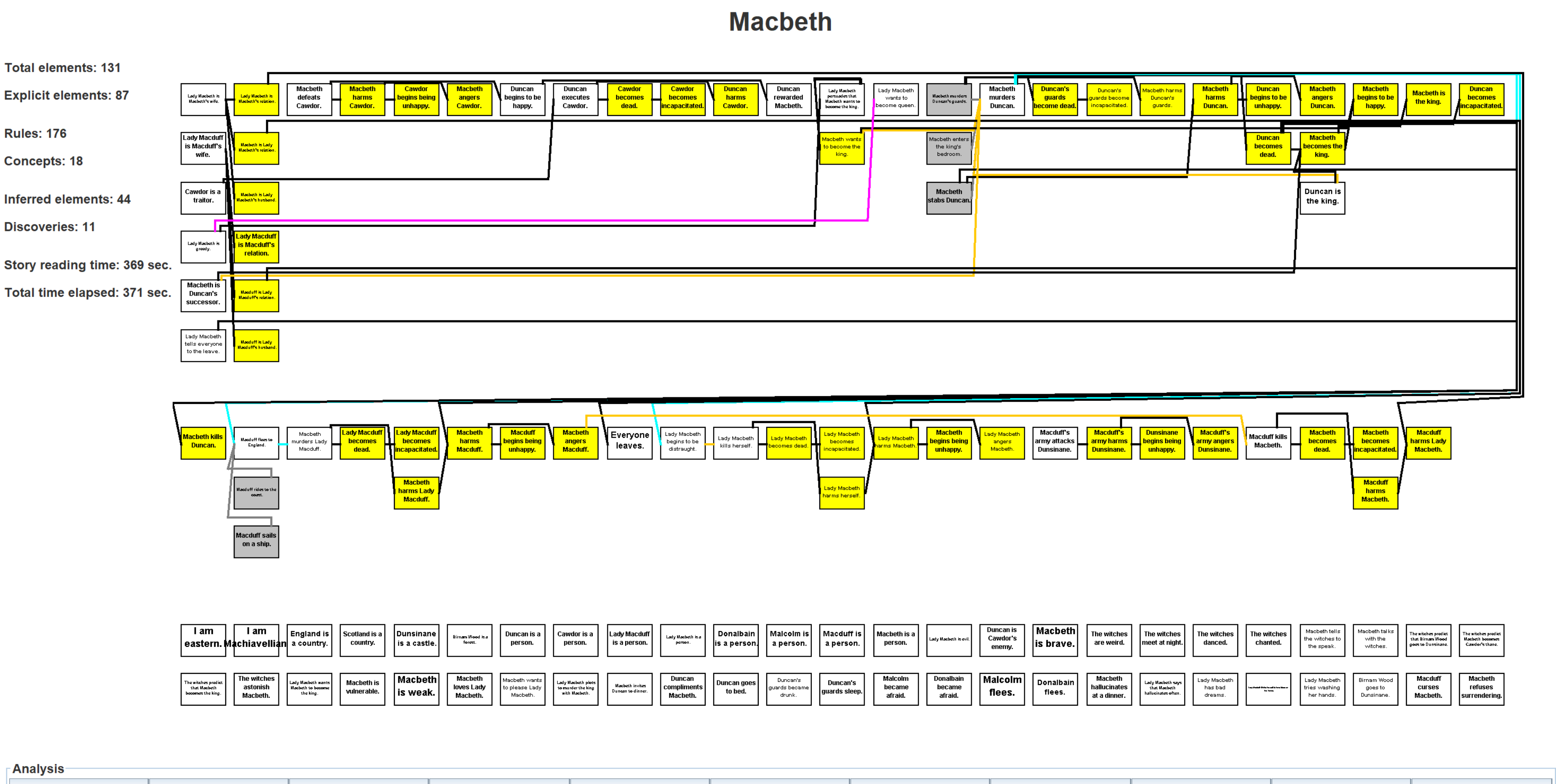


Figure A-2: Genesis produces elaboration graphs as shown for the Macbeth story. Elements in white are provided explicitly in the story. Elements in yellow are established by inference reflexes. Black connections represent deductions. Orange connections are inserted by explanation rules and cyan connections reflect sentences in the story that contain leads-to relations, both explained further in [2]. Note that although the story is told as a sequence of elements, the inference reflexes form long-distance connections.

Concept patterns are mini stories that Genesis can identify within a larger story, and often contain a “leads-to” relations, such as the concept of Revenge demonstrated in Figure A-3. Concept patterns are meant to represent broader themes or ideas in the story, and thus are useful as a comparative measure between stories. OPERA uses concept patterns specifically as part of its character alignment implementations. Detecting rules and concept patterns relies on Genesis’s matcher.

```
Start description of "Revenge”.
Xx and yy are entities.
XX’s harming yy leads to yy’s harming xx.
The end.
```

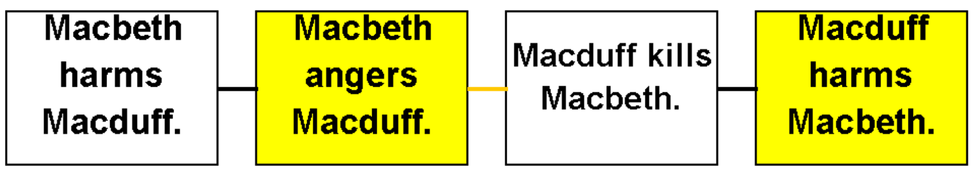


Figure A-3: Concept patterns are small stories that Genesis can identify within a larger story. Concept patterns often have a leads-to clause, requiring Genesis to perform search within a story to determine if there is a match between the story and the concept. Here we see the results of the Revenge concept from Genesis’s elaboration graph in A-2.

Genesis’s matcher determines if the structure of two entities align and if a successful matching is found, produces a set of bindings that map between the two entities’ corresponding elements. For example, consider two Role Frames translated from innerese to English: “John loves Mary,” “John loves Susan.” When provided with the translated innerese version of these two sentences, Genesis’s matcher would determine that the two entities match and would map John to John and Mary to Susan. OPERA uses a version of this matcher, augmented by Fay’s work with plot alignment, to generate all the bindings during character alignment.

B Stories Summaries

B.1 Anastasia

Anastasia is the princess. Marie lives in Paris. Anastasia lives in the palace. Anastasia's family lives in the palace. Marie gives Anastasia a music box. Rasputin was a royal advisor. Rasputin is an evil sorcerer. Anastasia's family exiles Rasputin. Rasputin trades Rasputin's soul for the evil talisman because Rasputin wants to harm Anastasia's family. Rasputin trades Rasputin's soul for the evil talisman because Rasputin wants to harm Anastasia. The Russian revolution starts because Rasputin curses Anastasia's family. The palace is burned because the Russian revolution starts. Rasputin thinks Anastasia is dead. Anastasia's family dies in the fire. Marie escapes from the palace with Dimitri's help. Anastasia escapes from the palace with Dimitri's help. Rasputin is stuck in limbo. Marie and Anastasia board a train to Paris. Anastasia falls off the train. Anastasia hits her head. Anastasia gets amnesia. 10 years pass. Marie posts a reward for the princess. Rasputin escapes limbo. Rasputin wants to kill Anastasia. Anastasia wants to go to Paris. Anastasia meets Vladimir and Dimitri. Vladimir thinks Anastasia resembles the princess. Dimitri thinks Anastasia resembles the princess. Dimitri wants to trick Marie with Anastasia. Vladimir wants to trick Marie with Anastasia. Dimitri and Vladimir take Anastasia to Paris. Rasputin orders Bartok to kill Anastasia. Bartok tries to kill Anastasia. Dimitri saves Anastasia from Bartok. Anastasia falls in love with Dimitri. Sophie works for Marie. Sophie believes that Anastasia is the princess because Sophie interrogates Anastasia. Marie does not believe that Anastasia is the princess so Marie refuses to see Anastasia. Dimitri kidnaps Marie because Marie refuses to see Anastasia. Marie does not recognize Dimitri. Dimitri shows Marie Anastasia's music box, so Marie agrees to see Anastasia. Anastasia remembers Marie. Marie believes Anastasia is the princess. Anastasia is reunited with Marie. Marie remembers Dimitri. Marie gives Dimitri the reward money. Dimitri refuses the reward money. Rasputin lures Anastasia to the forest in order to attack her. Anastasia remembers Rasputin. Dimitri defends Anastasia against Rasputin. Dimitri attacks Rasputin. Dimitri becomes unconscious because Rasputin attacks him. Anastasia destroys the evil talisman because Anastasia knows Rasputin harmed her family. Anastasia marries Dimitri. Anastasia and Dimitri live happily-ever-after. The end.

B.2 Hamlet

Ophelia loves Hamlet. Claudius wants to become the king. Claudius murders King Hamlet. Claudius marries Gertrude. The ghost tells Hamlet that Claudius killed King Hamlet. Hamlet believes Claudius killed King Hamlet. Hamlet dislikes Claudius. Hamlet wants to harm Claudius. In order to prove Claudius' guilt, Hamlet organizes a play. Hamlet believes Claudius is guilty. In order to confront Claudius, Hamlet goes to Claudius's room. Claudius is praying. Hamlet will not kill Claudius while Claudius is praying. In order to confront Gertrude, Hamlet goes to Gertrude's room. Polonius goes behind a curtain. Hamlet believes Claudius is behind the curtain. Hamlet stabs the person behind the curtain. Hamlet kills Polonius. Polonius becomes dead. Claudius sends Hamlet to England. Ophelia becomes insane. Ophelia kills herself. Hamlet returns from England.

Claudius poisons the sword. Claudius poisons the goblets. Laertes fights Hamlet. Gertrude drinks poison. Gertrude becomes dead. Laertes stabs Hamlet. Hamlet stabs Laertes. Laertes becomes dead. Hamlet stabs Claudius. Hamlet forces Claudius to drink poison. Claudius becomes dead. Hamlet becomes dead. The end.

B.3 Julius Caesar

Cassius wanted Caesar to die because Cassius hates Caesar. Cassius persuades Brutus to murder Caesar. Cassius also murders Caesar because Cassius persuades Brutus to murder Caesar. Anthony is Caesar's friend. Caesar is Anthony's friend. Anthony persuades the people to attack Cassius. Anthony persuaded the people to attack Brutus. Brutus and Cassius fight Anthony. Brutus kills Brutus. Cassius kills Cassius. The end.

B.4 Lion King

Mufasa, Simba, and Nala live in Pridelands. Simba is a child. Mufasa is the king and Simba is Mufasa's successor and Scar is Simba's successor. Scar is evil and greedy. Scar wants to become king. Scar is leader of the hyenas. Scar lures Simba to a gorge. Scare orders the hyenas to start a stampede. Scar tells Mufasa that Simba is in danger. Mufasa goes to the gorge. Mufasa saves Simba. Mufasa puts Simba on a ledge. Mufasa is hanging on the ledge because Mufasa slips. In order to murder Mufasa, Scar makes Mufasa let go of the ledge. Simba does not know that Scar killed Mufasa. Simba feels responsible for Mufasa's death. Scar persuades Simba to run away. Scar orders the hyenas to follow Simba and to kill Simba. Simba escapes the hyenas. Simba collapses in the desert. Timon finds Simba. Pumba finds Simba. Timon raises Simba in the jungle. Pumba raises Simba in the jungle. Scar tells the Pride that Mufasa died. Scar tells the Pride that Mufasa died. Many years pass. Simba is an adult. Nala attacks Timon. Nala attacks Pumba. Simba saves Timon. Simba saves Pumba. Nala recognizes Simba. Simba is reunited with Nala. Simba falls in love with Nala. Nala falls in love with Simba. Nala tells Simba that Scar is a dictator. Nala wants Simba to return to Pridelands. Simba does not want to return to Pridelands. Mufasa's ghost appears to Simba. Simba talks with Mufasa's ghost. Mufasa's ghost persuaded Simba to want to become the king. Mufasa's ghost persuaded Simba to return to Pridelands. Nala, Timon, and Pumba follow Simba. Simba confronts Scar because Simba wants to become king. Scar betrays the hyenas. The hyenas know Scar betrays them. ***Scar fights Simba. Scar confesses to Mufasa's murder. Simba fights Scar.*** ***Simba defeats Scar and Simba exiles Scar. The hyenas confront Scar. The hyenas kill Scar. Simba becomes king. Nala becomes the queen. Nala gives birth to Simba's son. Simba and Nala live happily-ever-after.*** The end.

****Text:*** text was removed for first part of example in experiment 5.2

****Text:*** text was added back for second part of example in experiment 5.2

B.5 Macbeth

Lady Macbeth is evil and greedy. Duncan is the king, and Macbeth is Duncan's successor. Duncan is an enemy of Cawdor. Macbeth is brave. Macbeth defeats Cawdor. Duncan becomes happy because Macbeth defeats Cawdor. The witches are weird. The witches meet at night. The witches danced and chanted. Macbeth tells witches to speak. Macbeth talks with the witches. Witches predict that Birnam Wood will go to Dunsinane. The witches predict that Macbeth will become Thane of Cawdor. The witches predict that Macbeth will become king. The witches astonish Macbeth. Duncan executes Cawdor because Cawdor is a traitor. Duncan rewarded Macbeth because Duncan became happy. Lady Macbeth wants Macbeth to become king. Macbeth is weak and vulnerable. Lady Macbeth persuades Macbeth to want to become the king because Lady Macbeth is greedy. Lady Macbeth wants to become queen. Macbeth loves Lady Macbeth. Macbeth wants to please lady Macbeth. Macbeth wants to become king because Lady Macbeth persuaded Macbeth to want to become the king. Lady Macbeth plots to murder the king with Macbeth. Macbeth invites Duncan to dinner. Duncan compliments Macbeth. Duncan goes to bed. Duncan's guards become drunk and sleep. In order to murder Duncan, Macbeth murders the guards, Macbeth enters the king's bedroom, and Macbeth stabs Duncan. Malcolm and Donalbain become afraid. Malcolm and Donalbain flee. Macbeth's murdering Duncan leads to Macduff's fleeing to England. In order to flee to England, Macduff rides to the coast and Macduff sails on a ship. Macduff's fleeing to England leads to Macbeth's murdering Lady Macduff. Macbeth hallucinates at a dinner. Lady Macbeth says he hallucinates often. Everyone leaves because Lady Macbeth tells everyone to leave. Macbeth's murdering Duncan leads to Lady Macbeth's becoming distraught. Lady Macbeth has bad dreams. Lady Macbeth thinks she has blood on her hands. Lady Macbeth tries to wash her hands. Lady Macbeth kills herself. Birnam Wood goes to Dunsinane. Macduff's army attacks Dunsinane. Macduff curses Macbeth. ***Macbeth refuses to surrender. Macduff kills Macbeth.*** The end.

****Text:*** text was removed for example in chapter 4 and experiment 5.1

C. Relevant Concepts

Answered prayer
`aa is an action.`
`xx's wanting aa leads to aa.`
Ex: Macbeth wanting to become king leads to Macbeth becoming king.

Avenge family
`yy is zz's relation.`
`xx's harming yy leads to zz's harming xx.`
Ex: Macbeth harming Lady Macduff, who is Macduff relation, leads to Macduff harming Macbeth.

Mistake because harmed
`aa is an action.`
`xx's wanting aa leads to yy's harming xx.`
Ex: Macbeth wanting to become king leads to Macduff harming Macbeth, so wanting to become king was a mistake for Macbeth.

Mistake because unhappy
`aa is an action.`
`xx's wanting aa leads to xx's becoming unhappy.`
Ex: Macbeth wanting to become king leads to Macbeth becoming unhappy, so wanting to become king was a mistake for Macbeth.

Pyrrhic victory
`aa is an action.`
`xx's wanting aa leads to xx's becoming happy.`
`xx's wanting aa leads to xx's becoming incapacitated.`
`xx becomes incapacitated after xx becomes happy.`
Ex: Macbeth wanting to become king leads to Macbeth becoming happy. However, Macbeth wanting to become king also leads to Macbeth becoming incapacitated when Macduff harms him. The victory (becoming king) makes Macbeth happy but is short lived because it eventually leads to hum becoming incapacitated.

Regicide
`xx is a king.`
`yy kills xx.`
Ex: Macbeth murders Duncan, who is the king.

Revenge
`xx's harming yy leads to yy's harming xx.`
`xx must not equal yy.`
Ex: Macduff harms Macbeth because Macbeth harmed Macduff.

Success
`aa is an action.`
`xx's wanting aa leads to aa.`
`aa leads to xx's becoming happy.`
Ex: Macbeth wants to become king leads to Macbeth becoming king. Macbeth is happy to be king and therefore was successful.

Suicide
`xx kills xx.`
Ex: Lady Macbeth kills herself.

References

[1] R. Berwick and N. Chomsky, Why Only Us: Language and Evolution, Cambridge: The MIT Press, 2015.

[2] P. Winston and D. Holmes, "The Genesis Enterprise: Taking artifical intelligence to the next Lleve via a computation account of human story understanding.," Computational Models of Human Intelligence (CMHI), Report Number 1, 2018.

[3] P. H. Winston, "The Strong Story Hypothesis and the Directed Perception Hypothesis," in *AAAI Fall Symposium: Advances in Cognitive Systems*, San Francisco, 2011.

[4] M. P. Fay, "Enabling Imagination through Story Alignment," MIT Department of Electrical Engineering and Computer Science, Cambridge, 2012.

[5] R. Hugh, G. Chiara and P. Martin, "Learning to predict or predicting to learn," *Language, Cognition, and Neuroscience,* vol. 31, no. 12, 2015.

[6] A. Bubic, D. Y. von Cramon and R. I. Schubotz, "Prediction, cognition, and the brain," *Fronteirs in human neuroscience,* vol. 4, no. 25, 2010.

[7] A. Clark, "Whatever next? Predictive brains, situated agents, and the future of cognitive science.," *Behavioral and Brain Sciences,* vol. 36, no. 3, pp. 181-204, 2013.

[8] M. Ramascar and M. Dye, "Error and expectation in language learning: The curious absence of "mouses" in adult speech," in *CogSci*, Amsterdam, 2009.

[9] Y. Nagai, "Predictive learning: its key role in early cognitive development.," *Philosophical transactions of the Royal Society of London. Series B, Biological sciences,* vol. 374, no. 1771, 2019.

[10] R. C. Schank, "Conceptual Dependency: {A} Theory of Natural Language," *Cognitive Psychology,* vol. 3, no. 4, pp. 532-631, 1972.

[11] N. Mostafazadeh, M. Roth, A. Louis, N. Chambers and J. Allen, "LSDSem 2017 Shared Task: The Story Cloze Test," in *EACL Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem)*, Valencia, 2017.

[12] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli and J. Allen, "A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories," in *ACL: Human Language Technologies*, San Diego, 2016.

[13] L. Deng and Y. Liu, Deep Learning in Natural Language Processing, Springer, 2018.

[14] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," vol. arXiv:1301.3781, 2013.

[15] L. Medrouk and A. Pappa, "Deep Learning Model for Sentiment Analysis in Multi-lingual Corpus," in *ICONIP*, 2017.

[16] T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," *CoRR,* vol. abs/1708.02709, 2017.

[17] J. Guan, Y. Wang and M. Huang, "Story Ending Generation with Incremental Encoding and Commonsense Knowledge," in *AAAI*, Honolulu, 2019.

[18] H. Liu and P. Singh, "ConceptNet — a practical commonsense reasoning tool-kit," *BT Technology Journal,* vol. 22, no. 4, p. 211–226, 2004.

[19] S. Chaturvedi, H. Peng and D. Roth, "Story Comprehension for Predicting What Happens Next," in *Conference on Empirical Methods in Natural Language Processing*, Copenhagen, 2017.

[20] J. Chen, J. Chen and Z. Yu, "Incorporating Structured Commonsense Knowledge in Story Completion," in *AAAI*, Honolulu, 2019.

[21] R. P. Hall, "Computational Approaches to Analogical Reasoning: A Comparative Analysis," *Artificial Intelligence,* vol. 39, no. 1, pp. 39-120, 1989.

[22] J. D. Becker, "The Modeling of Simple Analogic and Inductive Processes in a Semantic Memory System," in *IJCAI-69*, Washington DC, 1969.

[23] P. H. Winston, "Learning and reasoning by analogy," in *ACM*, New York, 1980.

[24] G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM,* vol. 38, no. 11, pp. 39-41, 1995.

[25] B. M. Williams, "A Commonsense Approach to Story Understanding," MIT Department of Electrical Engineering and Computer Science, Cambridge, 2017.

[26] B. Williams, H. Lieberman and P. H. Winston, "Understanding Stories with Large-Scale Common Sense," in *Thirteenth International Symposium on Commonsense Reasoning*, London, 2017.

[27] P. H. Winston, "Model-based Story Summary," in *6th International Workshop on Computational Models of Narrative*, Atlanta, 2015.

[28] F. G. Patterson, "The gestures of a gorilla: Language acquisition in another pongid," *Brain and Language,* vol. 5, no. 1, pp. 72-97, 1978.

[29] P. H. Winston, "The Genesis Story Understanding and Story Telling System: A Twenty-First Century Step toward Artificial Intelligence.," Memo 019, Center for Brains Minds and Machines, MIT, Cambridge, 2014.

[30] B. Katz, "Annotating the World Wide Web using Natural Language," in *RIAO Conference on Computer Assisted Information Searching on the Internet*, 1997.

[31] P. H. Winston, "Genesis implementation substrate," [Online]. Available: https://groups.csail.mit.edu/genesis/Documentation/frames.pd.

[32] R. Jackendoff, Semantics and Cognition, Cambridge: MIT Press, 1983.

[33] C. Martin, F. Branzi and M. Bar, "Prediction is Production: The missing link between language production and comprehension," *Nature Scientific Reports,* vol. 8, no. 1079, 2018.

[34] D. Genter and K. J. Holyoak, "Reasoning and Learning by Analogy," *American Psychologist,* vol. 52, no. 1, pp. 32-24, 1997.