Automatic Extraction of Conceptual Relations from Children's Stories

A Final Thesis
Presented to
the Faculty of the College of Computer Studies
De La Salle University Manila

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Computer Science

by

SAMSON, Briane Paul V.

Ethel ONG
Adviser

April 17, 2013

Abstract

People use storytelling as a natural and familiar means of conveying information and experience to each other. During this interchange, people understand each other because we rely on a large body of shared common sense knowledge. But computers do not share this knowledge, causing a barrier in human-computer interaction and in applications requiring computers to generate coherent text. To support this task, computers must be provided with a usable knowledge about the basic relationships between concepts that we find everyday in our world.

This research made use of an existing software tool and custom extraction rules that will automatically extract concepts and their relations from existing children's stories, and store these in a knowledge base that Picture Books and other NLP applications can utilize to do their tasks. Based on the results of the evaluations, the extractor has been found to be inaccurate in identifying relations in a story. The incomplete and generalized templates, insufficient indicators, accuracy of existing tools, and inability to infer and detect implied relations were the main causes of inaccuracy. Furthermore, the quality and accuracy of extracted relations decrease as the complexity and length of a story increases.

Keywords: language parsing and understanding, text analysis, semantic networks, natural language processing

Contents

List of Figures

List of Tables

1 Research Description

This chapter discusses an overview of the current state of technology, the research objectives, scope and limitations, and its significance.

1.1 Overview of the Current State of Technology

Natural language processing systems use a set of knowledge base in order to do tasks such as text generation. But simple lexicons and large unstructured corpora may be insufficient as knowledge base of these systems. Storytelling, for instance, is a natural task for humans. Armed with a library of words, their meanings and their relationships, we combine words and events to tell stories about ourselves, our community, and our experiences. Thus computers must be provided with the same shared collection of common sense knowledge about the basic relationships between things and events that nearly every person knows in order for them to achieve a level of expressiveness same as humans and be able to understand the world that we talk about. Such knowledge are represented as conceptual relations defining the relationship between two or more concepts in real life.

Recent creative text generation systems such as (Hong & Ong, 2009) have utilized a semantic network representation of concepts on common sense knowledge to identify relationships of words in human puns in order to generate computer puns. Another system, Picture Books (Solis, Siy, Tabirao, & Ong, 2009), generates stories with morals for children ages 4 to 6, by using a semantic ontology, patterned after ConceptNet (Liu & Singh, 2004a), containing conceptual knowledge about objects, activities, and their relationships in a child's daily life. The process of building and populating the Picture Books ontology required a lot of manual effort on the part of the proponents. Currently, the ontology contains 240 concepts and 369 relations, which were populated based on the themes that have been identified as relevant for the target age group.

Early Information Extraction (IE) systems have addressed the extraction of information from relatively small collections of well-structured documents such as newswire or scientific publications (Muslea, 1999). More recently, IE systems are focused on extracting facts from structured and unstructured documents for a particular domain, such as legal documents (Cheng, Cua, Tan, & Yao, 2008).

Although IE systems are capable of recognizing entities within documents (e.g. 'Renoir' is a 'Person', '25 Feb 1841' is a 'Date'), the relation between the entities (e.g., 'Renoir' was born on '25 Feb 1841') is not extracted, thus generating incomplete information that may be needed by certain applications (Banko & Etzioni, 2008). A variant of IE, Relation Extraction (RE), is the task of recognizing the assertion of a particular relationship between two or more entities in text.

The task of relation extraction is difficult, but relations such as hypernymy (IsA) and meronymy (PartOf) are often expressed using a small number of lexico-syntactic patterns (Hearst, 1992). Using a sample set of 500 sentences selected at random from an IE training corpus, Banko and Etzioni (2008) showed that many binary relationships are also consistently expressed using a compact set of relation-independent lexico-syntactic patterns.

The Artequakt project (Alani et al., 2003) also showed that it is possible to automatically acquire such relations from documents to populate an ontology. Working in the domain of artists, the Artequakt project identifies relations between entities of interest within sentences, following ontology relation declarations and lexical information. These relations are then used to populate an ontology with knowledge triples for use in the generation of biographies of artists.

To convey the ideas of a story, Nakasone and Ishizuka (2006) developed a storytelling ontology model by identifying relations between sentences in the story using the Rhetorical Structure Theory (RST) of Mann and Thompson (1987). As Knott and Dale (1994) pointed out, explicit and implicit relations hold between the sentences of a text, so that the content of one sentence might provide justification, elaboration or explanation for the content of another. These relations bind a text together to contribute to the overall comprehension of a story by the readers; for instance, whether understanding one text span (scene of a story) increases the reader's readiness to understand another scene, or whether understanding both spans allows the reader to recognize a particular semantic relation as holding between them. Certain discourse relations or cue phrases, such as "but", "so", "although", "more precisely" or "for example", are used to signal explicit relations between text spans.

Although both IE and RE have achieved significant progress in extracting facts and concepts in the domains of newspapers (Muslea, 1999), biographies (Alani et al., 2003), and legal documents (Cheng et al., 2008), limited work has been done on children's stories. Furthermore, since stories contain sequences of actions that characters perform or experience at various points in the story world, knowledge about how these events are ordered and the constraints under which they can occur must also be extracted.

1.2 Research Objectives

1.2.1 General Objective

To develop a methodology that automatically identifies and extracts the relations between everyday concepts and objects from children's stories and store them in a semantic network to provide ontological knowledge for Picture Books.

1.2.2 Specific Objectives

1. To collect a corpus of children's stories;
2. To analyze the English sentence structures in the corpus;
3. To derive a set of extraction patterns;
4. To develop a representation for modelling relations of every object common in children's stories;
5. To implement extraction rules using GATE for extracting conceptual relations automatically from the corpus, and;
6. To validate the resulting conceptual relations extraction tool through comparison with a gold standard and integration with Picture Books

1.3 Scope and Limitations of the Research

At least 30 children's stories were collected to form the input corpus for the extraction tool developed. These were then modified to remove the dialogues and to clean of inconsistencies. English sentence structures in these stories were analyzed to identify the types of relations that are present. This information was then used to derive a set of patterns or templates for extracting conceptual relations. An existing extraction tool was used in utilizing the extraction patterns to automatically locate instances of a known relation in the corpus.

In terms of extraction patterns, one relation can be expressed in various ways in text. Consider the hypernymy (IsA) relation, wherein the following sentences are possible ways of expressing it:

The dog is a canine.
The dog is a kind of canine.

The dog, a canine, is
The dog is a type of canine.

But although lexico-syntactic extraction patterns can be mapped directly to relations, certain English sentence constructs require further analysis and decomposition in order to derive their corresponding relations. These include sentence structures containing conjunctions and embedded clauses, as shown in the examples below:

Cake is made of flour, sugar, and butter.
The boy is singing and the girl is dancing.
Anna, who is the queen, went to the market, while the king went to the mall.

In line with the objectives of this research, all simple lexico-syntactic extraction patterns were handled given that they are from ConceptNet and the corpora at hand. The more complex sentences were manually modified into simpler sentences. Only then that the relations were extracted.

The extracted knowledge was stored in a representation model that can be used by NLP systems, in this case, the Picture Books story generator. It was patterned after the design of the Picture Books ontology, which is also patterned after the design of ConceptNet, to validate the presence of the appropriate relations against those identified from the collected corpus.

Since stories are sequences of events, their analysis necessitate the creation of new relations to represent sequences of events, temporal relations between events, as well as the constraints under which certain events may take place. For example, during testing, evaluators noticed that one of the generated stories of Picture Books occurred at an inappropriate time; specifically, the first segment of the story that introduces the day, the place, and the main character, contained the following text:

The evening was warm. Ellen the elephant was at the school. She went with Mommy Edna to the school.

Since the knowledge base of Picture Books currently does not provide relations about when certain events can occur, the main character went to school in the evening.

Sixteen (16) new semantic relations, resulting from analyzing the sample corpus, and from reviewing other works such as (Mueller, 2003) for modeling time and event occurrences, were created in this research. The main relations, mostly from ConcepNet and Picture Books, include IsA, PropertyOf, PartOf, MadeOf, EventForGoalEvent, EventForGoalState, EffectOf, EffectOfIsState, CapableOf, OftenNear, LocationOf and, UsedFor. Additional relations include Happens(e, t) which represents that a fluent f holds at time t, HasRole which represent character roles, RoleResponsibleFor which represent actions/events done by a specific role and lastly, Owns which represent ownership.

Alani et al. (2003) noted that it is inevitable for duplicate and contradictory information to be extracted from the input corpus. But he further noted that handling such information is challenging for automatic extraction and ontology population approaches. In this research, only duplicate extractions were handled.

For evaluation purposes, the Picture Books' code base was not modified in any way to accommodate the additional relations and their corresponding entries in the ontology. The same goes for additional entries for existing Picture Books' relations. All modifications in Picture Books were done only in its database.

Mueller (1999) also noted that "story understanding goes beyond generating parse trees, disambiguating words, or filling templates, and includes the ability to answer arbitrary questions, generate paraphrases and summaries, fill arbitrary templates, make inferences, reason about the story, follow reasoning in the story, relate the story to general knowledge, and hypothesize alternative versions of the story." Thus, aside from having a huge collection of common sense knowledge, a computer system must also be able to "make inferences about states and events not explicitly described in the text" (Mueller, 2003), by performing common sense reasoning using knowledge about the world. This requires a multi-representational model of this knowledge for the various realms of space, time, needs and feelings to be built, and is beyond the scope of the current proposal.

Manual validation with the help of a linguist was not utilized. Instead, a gold standard was used to evaluate the extracted relations' accuracy and completeness. Aside from this, the extracted relations were automatically validated through the generated stories of Picture Books.

1.4 Significance of the Research

Researches in the field of natural language processing (NLP) seek to find ways to make human-computer interaction more fluent. But human-computer communication is hampered by the lack of a shared collection of common sense knowledge that people rely on when they communicate in order to understand each other. In order to make computers achieve the same level of expressiveness as humans, we must give them "a common language with richness that more closely approaches that of the human language" (Niles & Pease, 2001).

Although dedicated IE systems have been developed to extract information from various domains, this research is a first step towards extracting relations from children's stories. Storytelling is a natural and familiar means of conveying information and experience to listeners (Nakasone & Ishizuka, 2006), thus justifying the selection of this domain for the proposed research.

The knowledge base derived from extracting entities, concepts, and events can then be used for various applications. One such application is in story generation. Picture Books can use the knowledge base to generate more variants as well as longer stories. Currently, Picture Books generates stories containing 24 to 30 sentences, which not only vary according to the age of the reader, but is also dependent on the available relations between two concepts. Story generating applications can in turn be used for both educational and entertainment purposes.

In education, Riedl and Young (2004) applied narrative generation techniques to generate historical fictions for teaching history, which they defined as "the chronological record of significant events". Lester et al. (2007) explored integrating narratives into learning environments that teach microbiology to provide an "adaptive, effective pedagogy that is both motivating and meaningful".

In entertainment, story generation is applied to develop interactive fiction systems. Montfort (2009) defines interactive fiction as "a venerable thread of creative computing and a literary art". His Curveship project uses NLP techniques to create narratives in the virtual world, where the user directs the possible flow of the story. For his knowledge base, Montfort utilized a tree representation that describes the possible sequences of events and the relationship of events to one another, as well as models of objects in the virtual world. A similar system in the game area, Faade (Mateas & Stern, 2003) is a 3D interactive drama that makes use of artificial intelligence techniques to allow players to interact with the characters in the story by playing as one of the characters and typing textual commands that affect the flow and the outcome of the game (story). Young (2008) is also exploring the development of computational models to generate narratives for 3D virtual game environments, which are being considered as alternative approach to promote learning.

Story understanding system can also benefit from using the knowledge base. Story understanding requires an enormous amount of common sense knowledge, thus the question and answering system of Mueller (2007) has a limited scope focusing on modeling the spatial and temporal aspects of narratives involving one or two characters dining in a restaurant. He employed a combined technique using IE to extract key information about dining episodes, and common sense reasoning to build models of the dining episodes. The model is limited to only a single spatial layout consisting of the street, the dining room, and the kitchen, and further work can be done to extract information about the spatial layout from the text, and use this to construct models of room-scale space.

1.5 Research Methodology

This section discusses the systematic approach that was performed in order to accomplish the objectives of this research. Despite the chronological order of each phase, some were performed in parallel while some were backtracked to address some issues. Documentation and regular consultations with the adviser and work group members were done throughout the course of the research.

1.5.1 Requirements Analysis

After an extensive analysis of existing relevant implementations and literature, the researcher first identified the different conceptual/semantic relations that can be applied to the children's story domain. Then, redundancies in the existing relations of Picture Books was noted. Afterwards, an open-source tool was identified to aid in the extraction of relations. Lastly, en evaluation criteria was identified to validate the resulting ontology.

1.5.2 Data Gathering

During this stage, the children's story corpus consisting of 30 physical copies of children's stories were gathered and digitally encoded. Modifications were done, such as transforming dialogues into declarative sentences, to address a different scenario. Please see Appendix C for a sample of this modification. Lastly, the extraction templates were aligned with ConceptNet, while some were created based on sentence structures in the gathered children's stories.

1.5.3 Architectural Design

In this stage, the different components of the open-source tool were identified to enable a thorough and complete extraction. This includes a part-of-speech tagger, named-entity identifier, and gazetteer, among others. Other resources to be utilized were also identified. Furthermore, the data structures which will represent the semantic relations in Picture Books will also be analyzed and designed.

1.5.4 Implementation

The final selection of the open-source tool components and their respective customizations were done at this stage. The primary focus was given to the finalization of extraction templates and their actual implementation into rules. Lastly, extracted relations were parsed and collated.

1.5.5 Testing

Testing was done to ensure the quality of the extracted relations. Unit testing for each component was performed. After doing so, integration testing was performed to verify that each component receives the correct input from the previous component and generates the appropriate result for use by subsequent components.

Lastly, the outputs of the system were mainly evaluated by generating stories out of Picture Books. These output stories are then put through the evaluation criteria identified during the requirements analysis phase.

1.5.6 Documentation

Throughout the entire project, documentation was done to track progress. This was also to ensure that any changes and implementations in the requirements of the study will be reflected in the documents.

1.6 Calendar of Activities

Tables 1.1, 1.2 and 1.3 shows a Gantt chart of the activities. Each bullet represents approximately one week worth of activity. The overlapping activities ensure that any omissions and modifications will be changed immediately.

Table 1.1: Timetable of Activities for the months of January to July 2010

Activities (2010)	Jan	Feb	Mar	Apr	May	Jun	Jul
Requirements Analysis	●●	●●●●	●●●●	●●●●			
Data Gathering			●●	●●●			
Architectural Design				●●●	●●●●	●●	
Implementation					●●	●●●●	●●●●
Testing							●●●●
Documentation	●●	●●●●	●●●●	●●●●	●●●●	●●●●	●●●●

Table 1.2: Timetable of Activities for the months of August to December 2010

Activities (2010)	Aug	Sep	Oct	Nov	Dec
Requirements Analysis					
Data Gathering					
Architectural Design					
Implementation	●●●●	●●●●	●●●●	●●●●	
Testing	●●●●	●●●●	●●●●	●●●●	
Documentation	●●●●	●●●●	●●●●	●●●●	●●●●

Table 1.3: Timetable of Activities for the months of January to April 2013

Activities (2010)	Jan	Feb	Mar	Apr
Requirements Analysis				
Data Gathering				
Architectural Design				
Implementation	●●●●	●●●●		
Testing	●●●●	●●●●	●●●●	●
Documentation	●●●●	●●●●	●●●●	●

2 Review of Related Literature

This chapter elaborates on the related works and relation extraction systems. It also discusses on the different sets of semantic relations used by past systems. Lastly, it compares a variety well-known existing knowledge representations.

2.1 Information and Relation Extraction Systems

Over the years, there has been an increasing amount of interest in the automatic detection of semantic relations, with the goal of making computers understand text. The earliest works are those of Hearst (1992) and Berland and Charniak (1999).

Marking the start of the automatic acquisition of relations, Hearst (1992) developed a method that automatically extracts hyponyms (IsA) from a wide variety of texts. One example of this can be seen in the phrase, *Rizzy, a dog*. It shows a hyponymy relation between the words *Rizzy* and *dog*. In extracting hyponymy relations, she used a set of frequently occurring domain-independent lexico-syntactic patterns which undoubtedly define a hyponymy relationship. Though her method has shown encouraging results, it still had some drawbacks such as the ambiguity of some relations extracted. Because her patterns were based on sample sentences in the corpora and aimed to cover as much instances of the hyponymy relation as possible, some of the outputs were indicative of other types of relation. Lastly, she went on to suggest that her method can be used to automatically acquire other types of relation such as meronymy (PartOf).

Later that decade, Berland and Charniak (1999) used a statistical approach to find meronymy (PartOf) relations from a very large corpus. As an example, the phrase *the plot of the story* signifies a meronymy relation between the words *plot* and *story*. In determining such a relation, they used a method similar to Hearst (1992) by also using a pre-defined set of frequently occurring lexico-syntactic patterns. But instead of producing tuples which signify the relation, they focused on producing an ordered list of possible parts given a list of six seed words representing whole objects. The list includes book, building, car, hospital, plant and school. The plant seed word was added to the list to see if the algorithm can identify correct parts despite the ambiguity in the sense of the word. This experiment yielded accuracies lower than the five other seed words. They used statistical metrics to produce the ordered list of possible parts. Though they have stated that their comparable success against Hearst (1992) was due to the large corpora that they used, they were still not able to maximize their corpora to their advantage due to the limited number of wholes and patterns used. They produced a list with an accuracy of 55% for the top 50 parts and 70% for the top 20 parts overall.

Despite their efforts, Hearst (1992) and Berland and Charniak (1999) were not able to address the problem of ambiguity in their patterns and outputs. Cases of ambiguity may occur for patterns signifying a number of semantic relations. For example, *the room of the house* shows a meronymy (PartOf) relation while *the room of the boy* does not. Fortunately, Badulescu et al. (2006) also observed this from both works thus using it as his motivation in employing another approach which automatically extracts PartOf relations.

In tackling PartOf relations, Badulescu et al. (2006) used a knowledge-intensive and supervised method in contrast to what has been used by Berland and Charniak (1999). They trained the algorithm with manually annotated set of positive (indicative of meronymy) and negative (not indicative of meronymy) training samples to produce a decision tree and a set of rules. Particularly, they used C4.5 decision tree learning to produce the rules. After training, they were able to produce a comprehensive set of classification rules to cover almost all subtypes of PartOf relations. They then tested the said rules using two corpora and had an overall average precision of 80.95% and recall of 75.91%.

In comparison, Berland and Charniak (1999) used a few number of words to represent whole entities which have identifiable parts in their very large corpus. In addition, they limited themselves to single word entities and concepts. Badulescu et al. (2006), on the other hand, used an approach which utilizes WordNet and NERD to determine single and multiple word concepts in perspective thus making his approach more general. Lastly, instead of determining the parts of a predefined whole, their work can determine if two noun concepts are indeed part of a PartOf relation through the use of their decision tree and classification rules. Badulescu et al. (2006) also tried to replicate the testing done by Berland and Charniak (1999) in their work but because the corpora used were different, the same conditions cannot be applied.

The aforementioned systems aimed to extract specific relations present in an English text. But such relations, IsA and PartOf, though can be easily extracted, are not the only conceptual relations there is. In lieu of this, several systems have already extracted facts and relations openly from plain-texts (Agichtein & Gravano, 2000) (Banko & Etzioni, 2008), web documents (Alani et al., 2003) (Yates et al., 2007), legal documents (Cheng et al., 2008) and newspapers (Muslea, 1999).

Snowball (Agichtein & Gravano, 2000), an open relation extraction system, employed a novel strategy in generating patterns and extracting relational tables from plain-text documents, specifically newspaper articles. A training phase is done with minimal training samples from human users. The seed patterns are then used to extract new patterns and relation tuples. As part of its extraction process, the system statistically evaluates the newly generated patterns and tuples and retains only the reliable ones in the new iteration. The large-scale evaluation provides Snowball with a methodology to produce high-quality patterns. However, the system can only produce relational tables involving named-entities accurately labeled by Alembic, a third-party named-entity tagger employed by Snowball. An example of a relational table would be for ORGANIZATION and LOCATION pairs. Such a table can contain the pairs *Microsoft-Edmond* and *Boeing-Seattle* which shows that the organizations *Microsoft* and *Boeing* can be found in *Edmond* and *Seattle*, respectively. Though it is only correct to extract such relations, there are still those which do not only involve a couple of named-entities. Relations involving world states like that between morning and go to school, clearly shows that a relation can also be between named-entities and phrases. This scenario poses another limitation of Snowball which is similar to (Berland & Charniak, 1999). Another shortcoming of Snowball would be that it can only extract relations between two named-entities which is not always the case for conceptual relations.

Taking a different path in relation extraction systems, the Artequakt project (Alani et al., 2003) focused on the domain of artists' biographies and extracted conceptual relations in order to automatically generate biographical accounts of artists. In comparison to previous systems, this one did not use any pre-determined extraction patterns per se and neither did it learn extraction patterns as a pre-process. Instead, the system just had a list of pre-determined ontology relations that it wants to extract along with its pair of concepts. In the whole process, the Artequakt project made use of third-party tools such as the Apple Pie Parser for syntactic analysis or part-of-speech tagging, GATE for entity recognition and WordNet to supplement GATE and to aid in actual relation extraction.

In extracting the relations, the unstructured web documents first goes through an entity recognition tool (GATE). WordNet is also used to supplement in case GATE fails to recognize any named-entity. The document then goes through the actual extraction phase wherein it gets decomposed into paragraphs and sentences. The part-of-speech of each word in a sentence is then labeled. After this, the main components of a sentence such as the subject, verb and object are identified. The system then uses the verb and entity pairs in each sentence and matches them with a corresponding ontology relation and concept pairs. In case of any linguistic variation, WordNet is used to increase the chance of matching with ontology relations and concepts. In its initial experiment, 50 web documents describing 5 artists were used. Promising results were shown as the system was able to extract at most 3 thousand unique conceptual relations with 85% precision and 42% recall on the average. Its low average recall was due to the varying cardinality of some relations. A high recall is preferred for relations with multiple cardinalities like *places_visited* while high precision is more preferred for relations with a single cardinality like that of *birth_place*.

Though this work has driven away from the usual use of templates in order to extract their target relations, it still boasts of its portability. The use of ontology relations instead of painstakingly specifying every single template for each target relation takes away the need to force-fit a relation extraction system to a specific domain.

In 2007, Yates et al. was able to develop an open information extraction system named TextRunner. It processes a corpus of heterogeneous web documents in a single pass without any human intervention. Though this system does not focus heavily on solving the problems faced by previous systems like portability but rather focus on the scalability of RE systems to the web, its novel contributions can still be considered a solution to such problems.

In developing the system, Yates et al. (2007) used the problems of automation, corpus homogeneity and scalability as motivations. This led to the development of some novel components such as the single pass extractor, self-supervised classifier, synonym resolution and query interface. The single pass extractor tags the sentences with their part-of-speech tags and noun-phrase chunks. Through the self-supervised classifier, it then checks for every pair of noun phrases that are not too far apart and determines whether or not there is a relationship between them. But before this can be done, the classifier has to be trained with positive and negative samples before it can accurately decide which among the noun phrase pairs has a relationship.

Since TextRunner (Yates et al., 2007) does not have a pre-determined set of relations unlike previous works, there is a high chance that the system extracts different tuples representing only one relation. To solve this problem, the system used Resolver to cluster the extracted tuples into sets of synonymous relations and entities.

In evaluating the system, a corpus of 9 million web documents was used. And with that, TextRunner was able to extract approximately 7.8 million well-formed tuples. Human reviewers evaluated some 400 randomly selected extracted tuples and determined that they were 80.4% correct. The system was then further compared to the performance of another traditional IE system, KnowItAll. After using a set of ten high-frequency relations, there were more correct relations extracted by TextRunner than KnowItAll.

In trying to improve TextRunner (Yates et al., 2007), Banko and Etzioni (2008) developed new systems in order to conduct a survey on the differences of open and traditional relation extraction. In these systems, the Conditional Random Fields model was used to label instances of a relation between all possible entity pairs. This is already an improvement from the Nave-Bayes classifier used by TextRunner which chooses tokens between entities heuristically and only predicts whether these indicated a relationship or not. Conditional Random Fields, on the other hand, is an undirected graphical model used to model multiple interdependent variables.

O-CRF, the new open relation extraction system, performs a self-supervised training as with TextRunner. It uses independent heuristics and applies them to the PennTreebank in order to obtain labeled relational tuples which are then described with features. Such features include part-of-speech tags, regular expressions, context words and the combination of features six words to the left and six words to the right of the labeled word. The context words used here include only closed classes like prepositions and determiners. Function words like verbs and nouns are not utilized as context words. The labeled relational tuples are then used to train the CRF. In extracting relations, O-CRF first does a single pass over the corpus and uses phrase chunking to identify entities. The CRF is then used to identify and label the relations occurring between entity pairs. As with TextRunner, O-CRF is also beset with duplicate relations. This was solved by applying the Resolver algorithm to predict if two relation strings refers to the same thing.

In order to make comparisons, R1-CRF, a system applying the same CRF model was developed. But this time, the traditional relation extraction paradigm is utilized. Though the same graphical model is used, there were some tweaks in order to comply with the traditional paradigm. A relation is given in advance and instead of training the CRF unsupervised, hand-labeled positive and negative samples are used. And unlike O-CRF, R1-CRF can use context words besides closed classes.

After evaluation, O-CRF showed 88.3% precision and 45.2% recall. These show promising results in using open relation extraction. However, the usage of such a paradigm will only be essential if the number of relations is big or unknown. This is also essential for extraction jobs concerning massive corpora. On the other hand, traditional relation extraction is more suitable for extraction jobs with a small number of target relations.

2.2 Semantic Relations

The interest in the automatic extraction of semantic relations in text has become one of the growing interests among researchers in the NLP community. And in recent years, a number of them applied different classification techniques on various domains. This, however, led to a variety of disjoint classification schemes which later on became a nuisance to the advancement of the field.

Way back in 1987, Mann and Thompson (1987) presented Rhetoric Structure Theory which describes major features of the organization of natural text. This descriptive theory is used linguistically to characterize the structure of natural text in terms of relations between parts of the text. It is a hierarchical structure which identifies both the transition point of a relation and the items related. Though it can be used for large corpora, its scope is limited to monologues only. Dialogues and spoken text, which are present in stories, are not handled by RST.

The relations in RST are mainly classified into two: nuclear-satellite and multinuclear. The nuclear-satellite relations can still be further classified as presentational or subject matter relations. Presentational relations are those which aim to increase inclination in the reader. An example of this would the Evidence relation which aims to increase the belief of the reader on the nucleus of the relation. Other than that, Motivation, Justify, and Background, among others, are also considered as presentational relations. Subject matter relations, on the other hand, aims to make the reader recognize the relation. Such relations include Condition, Circumstance, Elaboration, Purpose and Volitional cause, among others.

Years after RST, Knott and Dale (1994) conceptualized a set of coherence relations. But instead of treating relations as constructs used to describe a text, relations were thought of as constructs with psychological reality. Using this as motivation, Knott and Dale (1994) developed a bottom-up methodology to define a set of relations using cue phrases which is a concrete linguistic indicator of a relation in a text. Unlike most theorists who define relations between entities in a sentence, the relations described in this work are mostly those between the sentences of a text, thus implicit in nature. Such coherence relations are sometimes made explicit through the use of cue phrases like *for example* and *before*. The relations based on the cue phrases are divided into seven classes, namely: sequence, situation, causal/purpose, similarity, contrast/violated expectation/choice, clarifying and interruption.

In the domain of medicine, Rosario and Hearst (2001) defined a classification scheme for two-word noun compounds. Though their data was from MedLine, a collection of biomedical journals, the classes and relations defined in the study was made as general as possible. To be more specific, there was more granularity than those in case frames but the relations were also more general than the ones classified in traditional information extraction systems. In their classification scheme, there were actually 38 relations divided into 12 classes. General relations are also mixed with domain-specific ones. Examples of general relations include time of, frequency, instrument, object, topic and location, among others while those domain-specific ones include defect in location, person/center that treats, defect, research on and bind, among others.

Rosario et al. (2002) continued the study on semantic relations for noun compounds. But this time, a different classification scheme was used. Instead of their previous two-level hierarchy, they used the MeSH hierarchy which is a multi-level lexical hierarchy of classifying relations for noun compounds with 15 classes at the topmost level. Each of the 15 topmost classes corresponds mainly to a specific medical terminology or field like Anatomy, Biology, etc. This scheme presents classes which are more granular and more specific to the medicine field.

In 2003, Nastase and Szpakowicz presented a classification scheme for noun-modifier pairs in base noun phrases. This scheme is a two-level hierarchy classification of semantic relations for noun-modifier pairs. The hierarchy has 5 top-level classes and 30 bottom-level classes. Its 5 superclasses include causality, temporality, spatial, participant, and quality. Causality relations are mainly those which show cause-effect relations. For example, the base noun phrase cold virus will have a cause relationship between them since the head word virus caused the modifier cold. But other than the usual cause and effect relations, there is also the purpose relation which exists whenever the head word is meant for the modifier. Such is the case for the base noun phrase concert ground where the head word ground has the purpose of having a concert. Temporality relations, on the other hand, express time. One example is the frequency relation which holds whenever the head word occurs every time the modifier occurs. This is evident in the base noun phrase weekly mass. Spatial relations pertain to having the nature of space. Such is the case for outgoing call which shows a direction relation. Participant relations, unlike previous superclasses, include relations similar to semantic roles. One example of this would be the agent role which exists when the modifier performs the head word. The base noun phrase fan boycott signifies such a relation since fan performs the boycott. Lastly, the quality relations are those specifying content, manner and type, among others.

The same year, Alani et al. (2003) used a classification scheme very specific to the domain of artists' biographies. The ontology was derived from the CIDOC Conceptual Reference Model ontology and further modified by adding classes and relations needed to represent pieces of information appropriate for artists. Examples of such relations include date of birth, place of birth and inspired by, among others. These ontology relations are then utilized in generating artist biographies.

Instead of concentrating on classifying semantic relations for noun compounds or base noun phrases, Moldovan et al. (2004) specified a scheme in classifying relations for a range of phrases. This includes 35 classes of relations spanning at various syntactic levels. They were mostly derived from the list of relations specified in previous researches. However, it only contains the most frequently used relations in a large corpus. Some of the relations include possession, temporal, part-whole, is-a, cause, purpose, frequency, stimulus, manner and location, among others.

Concentrating more on the field of story generation, Nakasone and Ishizuka (2006) developed a storytelling ontology model using RST (Mann & Thompson, 1987). The ontology was made as generic as possible since most storytelling ontology models were defined and constrained by the way the events were linked and the nature of the narratives. Instead of constraining the model with such notions, the solution was more focused on how the narratives were organized and communicated to readers. Since the domain of the model is story generation, the ideas and events are to be focused on the concept of a conflict. Hence, the RST relations utilized were categorized into two: Conflict or Resolution relations. Conflict relations describe how the current state of the story is changed. Such relations include Contrast, Solutionhood, Elaboration, Consequence and Sequence. Resolution relations, on the other hand, describe how to understand the current state of the story. Examples of this type of relation include Background, Cause, Purpose and Result, among others.

And just recently, Hendrickx et al. (2009) developed a system which does a multi-way classification of semantic relations between a pair of nominals. But this time, instead of classifying all possible semantic relations, the focus was just on nine mutually exclusive domain-independent semantic relations with enough exhaustive coverage. The list includes Entity-Destination, Instrument-Agency, Product-Producer, Content-Container, Component-Whole, Entity-Origin, Cause-Effect, Member-Collection and Communication-Topic.

2.3 Knowledge Representations

Common sense knowledge acquisition is not new in the Natural Language Processing field. Over the years, several knowledge repositories or databases have been developed like WordNet, VerbNet, Cyc, FrameNet and ConceptNet. These repositories contain entries ranging from syntactic to semantic in nature. Though most, if not all, contain semantic relations, there are certainly differences on the relations they contain and how they are represented.

Begun in 1984, CYC aims to formalize common sense knowledge into a logical framework. It stores knowledge of every day concepts, objects and events in axioms. The assertions are both manually and automatically done by knowledge engineers at Cycorp assuming that they are already known in the world. In representing the assertions, a first-order predicate calculus, named CycL, with an extension of some second-order features is used. The knowledge base is partitioned into "microtheories" which are a bundle of assertions. Some microtheories are partitioned based on their common assumptions while some are partitioned based on a specific domain and level of detail. This mechanism allows Cyc to infer faster by focusing on a specific microtheory. Each time an inference is made, new assertions may be added into the knowledge base (source: cyc.com).

One of the forerunners and arguably the most popular among knowledge bases is WordNet. It is a general purpose semantic knowledge base started in 1985 at Princeton University. Its database consists of words, mostly nouns, verbs and adjectives. Each entry is structured into senses and associated using a small number of semantic relations such as the synonym, is-a and part-of relations. These relations are represented in WordNet as a semantic network with each word as a node and the relations as edges.

In 1998, Fillmore et al. (1998) developed FrameNet, a lexical resource containing frame-semantic descriptions of each English lexical item (noun, adjective and verb). The semantic domains that FrameNet covers are the following: health care, chance, perception, communication, transaction, time, space, body, motion, life stages, social context, emotion and cognition. The whole lexical database is composed of a lexicon, the frame database and the annotated example sentences. Each lexical entry contains some usual information like part-of-speech as well as formulas which describe how elements of a semantic frame can be recognized. FrameNet, as what was previously stated, also defines the argument structure of each entry in the lexicon through roles but instead of using case-roles or thematic roles, each argument is given a role name relative to a certain concept. The data structures used to represent the lexical entries along with their semantic frames were implemented using SGML.

VerbNet (Kipper, Dang, & Palmer, 2000) is another repository of semantic information but unlike WordNet, Cyc and ConceptNet, this repository is more focused on verbs and their semantics. It is primarily a verb lexicon using Levin verb classes to represent the lexical entries. As its semantic information, the lexical resource relates each verb's thematic roles and semantic predicates with syntactic frames and restrictions.

Though VerbNet has semantic information included in its verb lexical entries, it still differs from what WordNet, Cyc and ConceptNet has. The verb lexicon stores semantic roles and not semantic relations. Note that they are two different things though they are both semantic in nature. Semantic roles exist between a verb and its arguments while semantic relations may exist between any parts of speech.

Combining the structure of WordNet and the semantic richness of Cyc, ConceptNet (Liu & Singh, 2004b) is a large-scale common sense knowledge database aimed to optimize practical inferences over real-world texts. It adopted the semantic network knowledge representation of WordNet and included 17 additional relations such as EffectOf, SubEventOf and CapableOf. This will provide a richer semantic network compared to what WordNet already has. However, there are still differences on the relations they contain. In WordNet, relations are more formal and is assumed to always happen while in the case of ConceptNet, it relations are more informal and defeasible. This means that since ConceptNet is geared towards a more practical inference, its relations may not always happen. One example would be the part-of relation between dog and pet. A dog will always be a canine but not a pet.

Having a set of only 20 relations is not much of an advantage over Cyc since it provides more than 20 and with more detail. However, compared to the use of CycL as a knowledge base representation, ConceptNet's semantic network representation makes it easier to make practical inferences.

3 Theoretical Framework

This chapter discusses the theoretical framework through which the research problem is examined.

3.1 Children's Stories

Children's stories are a subset of general fictional literature that deals primarily with childhood. Main characters are usually children but may not always be human. In most cases, children's stories can be considered fables because of animal characters that talk and act. Furthermore, children's stories tends towards fantasy and is optimistic in nature (Nodelman, 2008).

In terms of structure, children's stories are mostly direct and simple. The vocabulary used can be too simplistic and the writing style uses too many short simple sentences. Actions are highly emphasized while psychological events are usually implied in narrations. As for its overall theme, children's stories are traditionally didactic in nature. In essence, children's stories are created to primarily educate children through repetition.

3.1.1 Events

For a piece of literature that has a series of events to be considered a story, events has to be related and consistent with each other. Relations between events can be signified in two ways: temporal succession and causality. Temporal succession or sequence through time is the relation between two events wherein *Event A* happens before *Event B*. On the other hand, causality means *Event B* happened as a result of *Event A*. This poses a stronger relation between events thus making it a vital component to consider a text a story.

Aside from an event's relation to another event, events can also be related to a setting or world state. For example, the action/event of *going to school* for children can only happen *in the morning*. Other examples include *wearing a coat in the winter*, *study in class*, and *sleep at home*.

Lastly, events can be categorized as voluntary and involuntary. Such voluntary events arise from the intentional doing of a character while involuntary events happen accidentally or because of natural causes.

3.1.2 Transition Words

Relation between events, whether temporal succession or causality, are usually signalled in a sentence or span of text by transition words. Tables 3.1, 3.2 and 3.3 shows the different transition words that can signify temporal succession, causality and effect, respectively (Ang, Antonio, Sanchez, & Yu, 2010).

Table 3.1: Transition words signalling time

Transition Words
After; after a while
Before
Currently
During
Eventually
First, Second
Finally
Immediately
Initially
Lastly
Later
Meanwhile
Next
Previously
Simultaneously
Suddenly
Then
While
Yesterday

Temporal succession transition words such as *afterwards*, *later*, and *before* may also suggest causality.

Table 3.2: Transition words signalling cause or reason

Transition Words
Because
Due to
For
As
Since

Table 3.3: Transition words signalling effect

Transition Words
As a result
Because
As a consequence
Consequently
Hence
So
For this reason
Therefore
Thus

3.2 Semantic ontology and Semantic relations

An ontology is an artifact with a set of representational primitives to model knowledge for a particular domain (Gruber, 2008). The representational primitives are classes or objects, attributes of the objects and relationship of each object. The design of the semantic ontology of Picture Books is patterned after ConceptNet (Liu & Singh, 2004a), a large-scale common sense knowledge base.

The nodes used by ConceptNet are of three general classes representing noun phrases, attributes, and activity phrases. A semantic relation connects two concepts while a semantic category classifies them. The semantic relations are binary relation types defined by Open Mind Commonsense project (Singh et al., 2002). Some of the ConceptNet relations are shown in Table 3.8.

Table 3.4: Sample semantic relations from ConceptNet (Liu & Singh, 2004b)

ConceptNet relations
IsA(dog, animal)
PropertyOf(apple, red)
PartOf(window, house)
MadeOf(sculpture, clay)
FirstSubeventOf(yawn, sleep)
EffectOf(become tired, sleepy)
CapableOf(ball, bounce)
LocationOf(seesaw, playground)
UsedFor(spoon, eat)

3.3 Picture Books

Picture Books is a creative text generation system aimed for kids 4-7 years old. The following sections detail the different components of Picture Books that allow it to generate stories.

3.3.1 Knowledge

Table 3.5: Excerpt from a story generated by Picture Books with corresponding conceptual knowledge

Line	Story Text	Conceptual Knowledge
1	Rizzy the rabbit was in the living room.	
2	She played near a lamp.	
		CapableOf(lamp, break) ***ConceptuallyRelatedTo***(break, break object)
3	Rizzy broke the lamp.	
		EffectOf(break object, be scared)
4	She was scared.	
	:	
5	Rizzy told Mommy Francine that Daniel the dog broke the lamp.	
	:	
		LastSubeventOf(break object, get punished)
6	He got punished.	
		LastSubeventOf(get punished, grounded) ***IsA***(grounded, punishment)
7	Mommy Francine told Daniel that he was grounded.	
		LastSubeventOf(grounded, cry)
8	He cried.	

Picture Books generates a story for a given input picture that contain a background selected by the user from the background library, as well as the character and object stickers placed onto the background. The ontology is used to derive relations between concepts, which refer to objects in the picture as well as the theme associated by the system through the background. An excerpt of a generated story and the corresponding conceptual knowledge used is shown in Table 3.5.

In line 1, the main character (*Rizzy the Rabbit*) and the setting (*living room*) were determined from the character sticker placed onto the selected background by the user. In line 2, the object (*lamp*) may or may not be in the picture, but included in the generated story based on the theme that is associated to the background. In this example, the theme is *being honest* through admitting your mistake (that is, the main character must not lie about breaking the lamp).

Access to the ontology is needed to derive events that can happen next in the story, as shown in line 3, and the effects of the resulting event, shown in line 4. Line 5 is the starting point of the rising action, where the main character misbehaves (*told a lie*) and the subsequent events and effects of the misbehavior. All the knowledge needed by Picture Books to do its task were manually encoded by the proponents into the system, based on the identified background and themes, which are appropriate to the target age group. The knowledge in ConceptNet cannot be used directly as these are not suitable for the users of Picture Books. Thus, only some of the ConceptNet knowledge as well as relations were used to build the ontology of Picture Books. Table 3.6 lists some of these relations defined in Picture Books following the form <relationship>(<concept1>, <concept2>).

Table 3.6: Semantic relationships adopted from ConceptNet (Liu & Singh, 2004b) with sample concepts of Picture Books

Semantic Category	Semantic Relationships
Things	***IsA***(headache, pain) ***PropertyOf***(apple, healthy) ***PartOf***(window, pane) ***MadeOf***(toy car, clay)
Events	***FirstSubeventOf***(tell bedtime story, sleep) ***EventForGoalEvent***(go to grocery store, buy food) ***EventForGoalState***(clean up, be neat) ***EventRequiresObject***(play, toy)
Actions	***EffectOf***(become dirty, itchy) ***EffectOfIsState***(make friends, friendship) ***CapableOf***(toy car, play)
Spatial	***OftenNear***(sailboat, water) ***LocationOf***(teacher, school)
Functions	***UsedFor***(thermometer, check temperature)

3.3.2 Story Elements

Aside from its ontology, Picture Books uses story elements to help it plan for a story to generate.

Theme

Themes dictate the flow of the different plots in the story. It plans the each story through four stages namely, problem, rising action, solution and climax. Each stage can have a number of plots assigned to it. But during story generation, it can only select one for each stage.

Story Plot

The story plots is designed to contain instructions for each stage of the theme. It is composed of at least two author goals. Each author goal is run during story generation.

Author Goal

Each author goal is equivalent to scene in the generated story. It has to have at least two character goals. At least one for the goal and another for its consequent actions. All character goals are processed during story generations.

Character Goal

Character goals signify the actions done by the characters in the story. It is also the abstract representation of a sentence in the final story generated. Whenever a character goal is called, additional parameters are added to convey a complete thought. Each parameter takes the following format: <attribute_name>:<value>. Attributes can be either of the following: *target*, *instrument*, *patiens* and *agens*.

There are four types of values that can be assigned to character goal attributes:

1. Word ID/s
2. Derivable values

3. Ontology accesses
4. Inner character goals

In an ontology access, the needed value is to be fetched from the ontology database. Each is started with ***onto*** followed by the semantic category of the relation that you want to access first. The complete format is as follows: %onto<Category>(<concept1>[,<concept2>])%. *Concept1* and *concept2* can either be Word IDs, derivable values or another ontology access. If an ontology path is desired, the second concept must be added to signify the end of the path. Here is an example of an ontology access:

Instrument:ontoAction(%object%)

In trying to pass an *instrument* attribute, the character goal accesses the ontology for an *Action* relation with the dragged object sticker as the parent concept. The immediate relation will be returned and used as attribute. Now, here is another example of an ontology access:

Target:ontoAction(%object%,%child

In this instance, the character goal is now trying to search for a path in ontology starting with the input object and ending with the main character of the story.

3.4 ConceptNet

ConceptNet (Liu & Singh, 2004b) is a large-scale common sense knowledge database aimed to optimize practical inferences over real-world texts. It adopted the semantic network knowledge representation of WordNet and included 17 additional relations such as EffectOf, SubEventOf and CapableOf. This will provide a richer semantic network compared to what WordNet already has. However, there are still differences on the relations they contain. In WordNet, relations are more formal and is assumed to always happen while in the case of ConceptNet, it relations are more informal and defeasible. This means that since ConceptNet is geared towards a more practical inference, its relations may not always happen. One example would be the part-of relation between dog and pet. A dog will always be a canine but not a pet.

The ConceptNet semantic network was populated with concepts and relations through a distributed solution of acquiring common sense knowledge from the public using a web-based data entry mechanism of the Open Mind Common Sense (OMCS) project (Singh et al., 2002). OMCS employs both semi-structured and free-form data entry approaches. The semi-structured approach utilizes extraction patterns commonly used by IE systems. Each extraction pattern or template has slots that users can fill-up, and is mapped directly to a relation.

Given the template "*<X> is a kind of <Y>*", the possible values for <**X**> and <**Y**> that users can provide and the corresponding hypernymy (IsA) relations that are acquired are shown in Table 3.7.

Table 3.7: Sample values to derive the hypernymy (***IsA***) relations

<X>	<Y>	Relations
Apple	Fruit	IsA(apple, fruit)
Ball	Toy	IsA(ball, toy)
Rose	Flower	IsA(rose, flower)

Table 3.8 shows the rest of the ConceptNet (Speer & Havasi, 2012) relations with their corresponding sentence patterns.

3.5 GATE

GATE (General Architecture for Textual Engineering) is a general-purpose infrastructure aimed for natural language software development. It also aims to reduce integration overheads. This is done through the provision of standard mechanisms of data communication for the the different software components. GATE also uses Java and XML as its platforms.

As a language engineering architecture, GATE provides processing resources with ANNIE as its main resource. ANNIE provides a set of reusable processing resources to facilitate language engineering tasks. It consists of the following resources: tokeniser, sentence splitter, POS tagger, gazetteer, finite state transducer or semantic tagger, orthomatcher and coreference resolver. The tokeniser splits a given text into simple tokens. The sentence splitter splits the text into sentences. The POS tagger tags each word or symbol with their specific part-of-speech tags. The gazetteer consists of lists like that of cities and organizations. It can also consist of lists of indicators, like titles and other designators. The orthomatcher performs coreference or entity tracking through the recognition of relations between entities. The coreference resolver detects identity relations between entities. Lastly, the semantic tagger consists of tailor-made rules written in

Table 3.8: ConceptNet Sentence Patterns

Relation	Sentence Pattern
IsA	*NP* is a kind of *NP*
UsedFor	*NP* is used for *VP*
HasA	*NP* has *NP*
CapableOf	*NP* can *VP*
Desires	*NP* wants to *VP*
CreatedBy	You make *NP* by *VP*
PartOf	*NP* is part of *NP*
Causes	The effect of *VP* is *NP/VP*
HasFirstSubevent	The first thing you do when you *VP* is *NP/VP*
AtLocation	Somewhere *NP* can be is *NP*
HasProperty	*NP* is *AP*
LocateNear	You are likely to find *NP* near *NP*
DefinedAs	*NP* is defined as *NP*
SymbolOf	*NP* represents *NP*
ReceivesAction	*NP* can be *VP*
HasPrerequisite	*NP/VP* requires *NP/VP*
MotivatedByGoal	You would *VP* because you want *VP*
CausesDesire	*NP* would make you want to *VP*
MadeOf	*NP* is made of *NP*
HasSubevent	One of the things you do when you *VP* is *NP/VP*
HasLastSubevent	The last thing you do when you *VP* is *NP/VP*

JAPE language. These rules describe the patterns and annotations to be created. A JAPE grammar has a set of phases which consist of pattern rules. These phases run sequentially.

3.5.1 Document Reset

Since a single GATE document can go through a GATE application a number of times, the Document Reset resource must be included before any other resource. This will ensure that the document is returned to its original state, without all the previous annotations.

3.5.2 English Tokeniser

This GATE resource splits the input document into tokens such as words (uppercase vs lowercase), punctuations, symbols, spaces and numbers. After splitting the text, a transducer then improves the output of the normal tokeniser. The final output is based on the requirements of the Englsh part-of-speech tagger. A sample improvement is the joining of the three tokens *don*, *'* and *t* to produce two tokens *do* and *n't*.

3.5.3 Gazetteer

A gazetteer is a resource that identifies entity names within a text with based on predefined lists. Each gazetteer is a plain text file that has one entry per line. This may contain a set of names like names of characters, locations and job titles, to name a few. Shown below is an excerpt of the predefined gazetteer *jobtitles.lst*

account Manager
office Manager
analyst
engineer
sales manager
doctor
software engineer
business analyst
managing director

Because this resource can use a number of gazetteers to identify entity names, an index file (lists.def) is used. Each time this resource detects a gazetteer entry in the text, a *Lookup* annotation is created.

3.5.4 Sentence Splitter

The sentence splitter, as the name suggests, segments the input GATE document into sentences and creates a *Sentence* annotation. A sentence break is annotated as *Split*.

3.5.5 Part of Speech Tagger

This resource adds a part-of-speech tag as a feature to the existing Token annotations. It uses a default lexicon and ruleset. It was the result of training on a large dataset from the Wall Street Journal.

3.5.6 Semantic Tagger

Based on the JAPE language, the named-entity transducer resource uses the annotations created in previous phases to produce outputs of annotated entities. Sample annotations created in this phase are *Person* and *Location*.

3.5.7 Orthographic Coreferencer

The Orthomatcher is mainly responsible in making sure that named entities identified by the semantic tagger which mean the same thing are matched accordingly. It does not actually produce new annotations. Rather, this resource adds a *matches* attribute to the named entities which contain the IDs of matching entities. For example, if the words *Coca-Cola* and *Coke* were tagged in a text, the Orthomatcher will add the *matches* attribute containing the IDs of these two words.

As resource, it uses a lookup table of aliases to perform coreference.

3.5.8 Pronominal Coreference

In a nutshell, this GATE resource identifies the named entity a pronoun is referring to in a text. It uses the output of pervious phases, specially the Orthographic Coreferencer. And like the Orthomatcher, it just adds the *matches* attribute to the appropriate annotation.

3.5.9 Morphological Analyzer

The GATE Morphological Analyzer takes a tokenized document as input. It then runs through all the tokens and based on their part-of-speech tag, the lemma and affix are identified. These are then added as attributes to the token.

3.5.10 Chunker

Though not really a part of GATE, this additional resource by OpenNLP identifies the different phrases or *chunks* in sentence. For each token, a *chunk* attribute is added based on the *category* of a token. Here are some *chunks* identified by this resource:

B-NP token begins of a noun phrase;
I-NP token is inside a noun phrase;
B-VP token begins a verb phrase;
I-VP token is inside a verb phrase;
O token is outside any phrase;
B-PP token begins a prepositional phrase;
B-ADVP token begins an adverbial phrase.

3.5.11 Transducer

JAPE or Java Annotation Patterns Engine, allows the identification of regular expression in annotation on GATE documents. One must create a JAPE grammar that contains phases running sequentially. Each phase has a set of rules that run based on priorities. Each rule has a left-hand-side (LHS) and a right-hand-side (RHS). The LHS contain the annotation pattern while the RHS contain the action to be done once the pattern is matched. Shown below is a sample JAPE rule:

```
Phase: Jobtitle
Input: Lookup
Options: control = appelt debug = true
Rule: Jobtitle1
(
Lookup.majorType == jobtitle
(
Lookup.majorType == jobtitle
)?
)
:jobtitle
–¿
:jobtitle.JobTitle = rule = ”JobTitle1”
```

In this example, the LHS tells us that the rule is trying to look for 2 consecutive Lookup annotations with the *majortype* attribute equal to *jobtitle*. The second Lookup annotation is optional because of the question mark symbol (?) following it. After recognizing the said annotation pattern, the rule then creates a new annotation *JobTitle* and adds the *rule* attribute to it. It will have the value JobTitle1.

3.6 Template-Based Extraction

In extracting semantic relations, one technique is through the generation and use of extraction patterns. For each target relation, a set of extraction patterns are needed to handle all possible instances of a relation in a sentence.

Table 3.9 shows other extraction patterns and the corresponding relations of ConceptNet.

Table 3.9: Sample extraction patterns and corresponding ConceptNet relations

Extraction Pattern or Template	Relations
CAKE is a kind of FOOD.	IsA(cake, food)
CAKE is made of FLOUR.	MadeOf(cake, flour)
FLOUR is WHITE.	PropertyOf(flour, white)
The effect of DRINKING MILK is GOOD HEALTH.	EffectOf(drinking milk, good health)

From the examples above, an instance of an extraction pattern generates one relation. But sentences may contain conjunctive phrases, which in turn may result to multiple relations being learned, as shown in Table 3.10 for the pattern "*<X> is made of <Y>*".

Table 3.10: Generating multiple relations from a single extraction pattern

Extraction Pattern or Template	Relations
CAKE is made of FLOUR, SUGAR, and MILK.	MadeOf(cake, flour) MadeOf(cake, sugar) MadeOf(cake, milk)

Part-of-speech tags may also be utilized to identify phrases and its constituents. For example, in Table 3.11, the noun phrase used to fill the <*X*> variable in the *IsA* template has three components, namely an article ("*the*"), an adjective ("*sweet*"), and a noun ("*cake*"). Extracting this knowledge can lead to the relation *PropertyOf(cake, sweet).*

Table 3.11: Utilizing POS tags for implicit relations

Input Sentence following a Template	Relations
The sweet cake is a dessert.	Explicit extraction pattern: IsA(dessert, cake) Implicit from POS tag: PropertyOf(cake, sweet)

The input stories may contain complex sentence structures, such as conjunctions and embedded clauses. Consider the sentence "*Anna, who is the queen,* *went to the market; meanwhile, the king went to the mall.*" By identifying and transforming this to three simpler sentences: "*Anna is the queen. She went to the market. Meanwhile, the king went to the mall.*", the following relations can be extracted.

IsPerson(Anna)
HasRole(person, queen)
HasRole(person, king)
CapableOf(person, go)
TargetOf(go, market)
TargetOf(go, mall)

4 Design and Implementation

This chapter discusses the design and implementation of the open-source tool to extract relations. It includes the architectural design, extraction rules and issues encountered during development.

4.1 Corpus

The input corpus for this study is comprised of 30 children's stories. Each were encoded digitally and modified to remove dialogues. The raw version was retained and used with the modified stories. For this research, the 2 corpus are named *RAW* and *MODIFIED*. Overall, 60 stories were used to extract the relations. Shown in Table 4.1 are the different story groups and their corresponding age groups.

Table 4.1: Children's Story Groups

Group Name	No. of Stories	Age Group
Topsy Tim	5	4-7 y.o.
Little Life Lessons	16	4-7 y.o.
Jumpstart	7	8-10 y.o.
Winnie the Pooh	2	8-10 y.o.

4.1.1 Modifications

As mentioned, each children's story in the corpus were modified to clean the data of any inconsistencies. This section details the different modifications done on the corpora.

Dialogues

Stories are mostly composed of dialogues which are conversational in nature. They are mostly informal, has incomplete thought and use colloquial words. A dialogue has different elements, namely:

- **Quotation marks** - punctuation that signal a character's spoken word. It also defines the end of a narration and the start of the speech.

- **Speech** - these are the spoken words.
- **Speaker Attributions** - the combination of a verb and a speaker. This signals the character speaking and the manner a speech is spoken.

Each element will be handled differently depending on the type of modification. The objective of these modifications is to convert these dialogues into complete and coherent sentences in order to yield proper extractions. In order to illustrate the dialogue modifications, here is an unmodified excerpt from one of the stories already in the corpus entitled "A Wild Weather Day".

It was a wild and windy day. The JumpStart ship was headed for Tree Fort Island.
Frankie was at the wheel. The sails flapped in the wind. The ship raced through the water.
"Why are we going so fast?" Pierre asked. "The wind is blowing us along on our adventure," CJ said. "Did you know that wind is just air that is strong and fast?"
"Like Frankie!" Pierre said. "He's strong and fast, too."
"Why is the sky getting so dark?" Pierre asked.
"I know why it's dark!" Eleanor said. "Clouds get dark when they fill up with tiny drops of water."
"Look! They're almost the same color as your bow," Pierre said.
A big drop of rain fell on Pierre's nose. "Oh, no!" he said. "It's starting to rain!"
"The rain is coming from the clouds," CJ said.
"The water in the clouds got too heavy, and now it's falling down on us!"
"Just like when Hopsalot waters his garden," said Pierre.

The first type of modification is done by transforming the dialogues into declarative sentences. Here is the modified version of the excerpt above:

It was a wild and windy day. The JumpStart ship was headed for Tree Fort Island.
Frankie was at the wheel. The sails flapped in the wind. The ship raced through the water.
They are going so fast. The wind is blowing them along on their adventure. The wind is just air that is strong and fast.
Frankie is like the wind. He's strong and fast, too.

The sky is getting so dark.
Eleanor knows why it is dark. Clouds get dark when they fill up with tiny drops of water.
They are almost the same color as Eleanor's bow.
A big drop of rain fell on Pierre's nose. It is starting to rain!
The rain is coming from the clouds.
The water in the clouds got too heavy, and now it is falling down on them!
Just like when Hopsalot waters his garden.

The second type is the usual transformation of direct to indirect speech.

It was a wild and windy day. The JumpStart ship was headed for Tree Fort Island.
Frankie was at the wheel. The sails flapped in the wind. The ship raced through the water.
Pierre asked why they were going so fast. CJ said that the wind was blowing them along on their adventure. He also asked if they know that wind is just air that is strong and fast.
Pierre said like Frankie. Pierre said that Frankie was strong and fast, too.
Pierre asked why the sky was getting so dark.
Eleanor said that she knew why it is dark. She explained that clouds get dark when they fill up with tiny drops of water.
Pierre said look! They are almost the same color as Eleanor's bow.
A big drop of rain fell on Pierre's nose. He was shocked. It is starting to rain.
CJ said that the rain is coming from the clouds.
The water in the clouds got too heavy, and now it is falling down on them.
Pierre said that it is just like when Hopsalot waters his garden.

Through these examples, it is noticeable how one dialogue can be transformed two ways. In comparison, the first type of modification transforms dialogues into factual statements, sometimes even altering or removing the speaker attribution (verb and speaker). Here is a passage from the example:

"Like Frankie!" Pierre said. "He's strong and fast, too."

The verb *said* will be removed as well as the character (*Pierre*) saying the dialogue. Since the focus is mostly on what is being said, and not who said it and the manner of saying, it was transformed into the following:

Frankie is like the wind. He's strong and fast, too.

On the other hand, the second type of modification usually retains the speaker attribution in its transformations. However, there can be changes in the verb used. Only one of these modifications are applied for each dialogue. The following rules were followed in transforming dialogues:

1. For dialogues that are declarative in structure, the first type of modification is followed.

   "I know why it's dark!" Eleanor said.
   to
   Eleanor knows why it is dark.

2. For dialogues that are interrogative in structure and conveys a complete observation, action, or thought, the first type of modification is followed.

   "Why is the sky getting so dark?" Pierre asked.
   to
   The sky is getting so dark.

3. For dialogues that are interrogative in structure but has incomplete thought, action, or thought, the second type of modification is followed. These dialogues are usually follow up questions to previous statements made by other characters.

   "Why so?" Pierre asked.
   to
   Pierre asked why so.

4. For all dialogues transformed using the second type, the verb in the speaker attribute is changed to *asked* or *said*. Some examples include, *whispered*, *uttered* and *mumbled*. This is done since these are just modifiers to how the dialogue was spoken.

After doing these modifications, it is important to note that the story has already changed. But in essence, the theme is still there. The intention was to make the actions and facts more apparent to the extraction tool.

Interjections

While modifying, there may be cases wherein a specific sentence or a line uttered by a character will be omitted or transformed into a declarative sentence not containing the original word/s. Such sentences can be interjections like "Oh my!" and "Alas!". These can be transformed as "She was shocked" and "He was concerned." The emotions conveyed by these interjections were used.

Contractions and Periods

Aside from these, modifications such as removing contractions and periods (.) that do not actually mean the end of a sentence were done. With contractions, they were put back in their long form. For example, *they'll* was transformed to *they will*. This modification does not mean the tool used to extract the relations could not handle contractions. This just makes the results cleaner. It also disambiguates contractions from possessive nouns. For example, instances like *Helen's* may mean ownership or just *Helen is*. Removing nuances like these allows the tool to produce a more valid POS tagging and of course, extraction.

As with the periods (.) like those found in titles (*Mr.* and *Dr.*), they were removed and the titles were transformed back into their long forms. For example, *Mrs.* is transformed into *Missus*.

Non-English Words

Lastly, story-specific modifications were done for the two *Winnie the Pooh* stories. After careful examination, it was evident that the author was using non-existing English words in the dialogues of its characters. Words like *suspicionated* and *splendiferous* appeared in the story *Everyone is special*. *Tigger* is usually the character that uses such words due to his character trait. Such words were transformed to their actual English word counterparts. For example, *suspicionated* and *splendiferous* were transformed into *suspected* and *splendid*, respectively. This will make sure that all words are correctly tagged and the extracted relations contain valid concepts.

4.2 Target Relations

Table 4.2: Target Relations

Element	**Description**	**Example**
IsA	Specifies what kind an entity is.	IsA(dog, pet)
PropertyOf	Specifies an adjective to describe an entity.	PropertyOf(mango, yellow)
PartOf	Specifies the *parthood* of an entity in another entity.	PartOf(knob, door)
MadeOf	Specifies a component of an entity.	MadeOf(door, wood)
CapableOf	Specifies what an entity can do.	CapableOf(kid, jump)
OftenNear	Specifies an entity near another entity in most instances.	OftenNear(chair, table)
LocationOf	Specifies the location of an entity.	LocationOf(slide, playground)
UsedFor	Specifies the use of an object in an activity.	UsedFor(toy, play)
EventForGoalEvent	Represents an event that causes the fulfillment of a goal event.	EventForGoalEvent(go to playground, play football)
EventForGoalState	Represents an event that causes the fulfillment of a goal state.	EventForGoalState(take a bath, be clean)
EffectOf	Represents a cause-effect between two events.	EffectOf(skip breakfast, hungry)
EffectOfIsState	Represents a cause-effect between an event and an end state.	EffectOfIsState(eat vegetables, health)
Happens	Specifies the time an event/state happens.	Happens(breakfast, morning)
HasRole	Specifies the role on a person in the story.	HasRole(Kisha, teacher)
RoleResponsibleFor	Specifies an action done by a role.	RoleResponsibleFor(doctor, diagnose)
Owns	Specifies the ownership of an object.	Owns(teacher, book)

For the purpose of this research, sixteen (16) relations were identified to be extracted. They were deemed to be related and helpful to the development of common sense knowledge for the children's story domain. Table 4.2 contains the final list of target relations to be extracted in this study.

One of the identified limitations in the current knowledge base of Picture Books is the lack of relations to denote event occurrences. Consider the following text:

The evening was warm. Ellen the elephant was at the school. She went with Mommy Edna to the school.

If appropriate relations are available, e.g., *Happens* to designate that an activity, such as going to school, can only happen at a certain time of day, such as morning, then the resulting text can be:

The morning was sunny. Ellen the elephant was at the school. She went with Mommy Edna to the school.

Certain granularities can be provided to the relations representing various aspects of time, namely season (planting can only occur during spring, snow can only fall during winter), month (Christmas in December, Valentine's in February), or even weeks, days, hours, and minutes.

Based on their intended functions, these relations can be classified into different groups (See Table 4.3).

Table 4.3: Classification of Target Relations

Category	**Relations**
Things	IsA, PropertyOf, PartOf, MadeOf, HasRole
Agents	CapableOf, RoleResponsibleFor
Events	EventForGoalEvent, EventForGoal-State
Causal	EffectOf, EffectOfIsState
Spatial	LocationOf, OftenNear
Functional	UsedFor
World	Happens, Owns

The *things* category can be used to describe the characters, objects, events and settings of the story. The *agents* category specifies the intentional actions that can be done by the characters. *Events* describe the temporal succession of events in terms of desire while *causal* relations provide information on causality. *Spatial* relations describe the location of objects and characters. *Functional* describe the actions that can be done with an object. Lastly, *world* relations provide a universal truth about characters, objects and events.

4.3 Extraction Templates

Extraction templates refer to the different ways a certain relation is manifested in a sentence or a span of text. Since most of the relations to be extracted were adopted from ConceptNet, the initial decision was to just follow the templates they have been using to crowd-source data (see Section 3.4). However, due to the contrast of sentence complexity between ConceptNet sentence patterns and the sentences in the corpus, additional templates were added to address different instances.

Table 4.4: Template Elements

Element	**Description**
<NP>	Noun Phrase
<NP:JobTitle>	Noun Phrase indicating a Job Title
<Noun:Possessive>	Possessive Noun
<AP>	Adjective Phrase
<Pronoun:Possessive>	Possessive Pronoun
<Verb>	Verb in root form
<VP>	Verb Phrase
<VP:Gerund>	Gerund
<PP:Temporal>	Temporal Prepositional Phrase
<Event>	Event (usually a VP)
<GoalEvent>	A desired event
<GoalState>	A desired state
<Cause>	Cause
<Effect>	Event effect
<EffectState>	State effect
<Indicator>	Relation-specific Indicator

Table 4.4 shows the different elements present in an extraction template. Nine (9) are tagged chunk information while the rest are custom tags created for this study. Lastly, *indicators* (<Indicator>) are used to denote the presence of a relation in the sentence. This is similar to the transition words discussed in Section 3.1.2. These will be discussed in detail in Section 6.1.4.

Table 4.5: Extraction Templates of each Relation (within a sentence)

Relation	**Extraction Template/s**
IsA	<NP> <IsAIndicator> <NP> : A dog is a kind of canine. <NP>, <NP>, is : The dog, a canine, is...
PropertyOf	<AP> <NP> : The red ball... <NP> ... <AP> : The ball is red.
PartOf	<NP> <PartOfIndicator> <NP> : A window is a part of a house. <Noun:Possessive> ... <NP> : The house's window... <Pronoun:Possessive> ... <NP> : Her head...
MadeOf	<NP> <MadeOfIndicator> <NP> : A cake is made of flour.
OftenNear	<NP> <OftenNearIndicator> <NP> : The vase is near the window.
CapableOf	<NP> ... <Verb> : The boy jumps. <NP> <CapableOfIndicator> <VP> : The boy can jump.
LocationOf	<NP> <LocationOfIndicator> <NP> : The slide is at the playground.
UsedFor	<NP> <UsedForIndicator> <VP> : A rolling pin is used for baking. <VP:Gerund> <UsedForIndicator> <NP> : Baking requires a measuring cup. <VP> <UsedForIndicator> <NP>: Kisha hit with a bat.
Owns	<NP> <OwnsIndicator> <NP> : Kisha owns a car. <Noun:Possessive> ... <NP> : The dog's collar... <Pronoun:Possessive> ... <NP> : Their home...
Happens	<VP> ... <PP:Temporal> : ...go to school in the morning. <PP:Temporal> ... <VP> : At night, Kisha sleeps...
HasRole	<NP> <IsAIndicator> <NP:JobTitle> : Helen is a teacher. <NP>, <NP:JobTitle>, is : Helen, a teacher, is...
RoleResponsibleFor	<NP:JobTitle> ... <VP> : The doctor cleaned... <VP> by <NP:JobTitle> : Kisha's sickness was diagnosed by the doctor.

Shown in Table 4.5 are the templates used for relations that can be extracted within a single sentence. While in Table 4.6, shown are the templates used for relations that can be extracted within a sentence and across 2 sentences. Beside each template is a sample sentence. All templates are already the combination of ConceptNet sentence patterns and those manually derived from the corpus.

In all templates, compounds are handled accordingly and separate relations are created for each concept. And as a limitation of the PartOf template, when the first concept is a person or animal, the template just looks for a body part as the second concept.

Table 4.6: Extraction Templates of each Relation (across sentences)

Relation	**Extraction Template/s**
EventForGoalEvent	<GoalEvent> ... <Event> : Kisha wants to buy a car. She saved all her lunch money. <Event> <MotivationIndicator> <GoalEvent> : Kisha saved all her lunch money because she wants to buy a car.
EventForGoalState	<GoalState> ... <Event> : Kisha wants to be slim. She ran around the park during mornings. <Event> <MotivationIndicator> <GoalState> : Kisha always ran in the morning because she wants to be slim.
EffectOf	<Cause> ... <Effect> : Because of the accident, the child cried. <Effect> ... <Cause> : The child cried because of the accident.
EffectOfIsState	<Cause> ... <EffectState> : Because of the accident, the child was sad. <EffectState> ... <Cause> : The child was sad because of the accident.

Table 4.6 shows the relations that can be manifested within a span of 2 sentences. Here is an example from the table:

Kisha wants to buy a car. She saved all her lunch money.

The <GoalEvent> is in the first sentence while the <Event> is on the second sentence. Thus, the relation *EventForGoalEvent(buy a car,saved all her lunch money)* can be extracted. Though this is mostly and almost always true for these 4 relations (*EventForGoalEvent*, *EventForGoalState*, *EffectOf*, *EffectOfIsState*), same is also the case for some of the relations in Table 4.5. Here is an example for the MadeOf relation that spans 2 sentences:

Kisha will bake a cake. It is made of flour, butter and sugar.

In this particular example, the indicator *is made of* is present in the second sentence. And since there is only 1 object that the *it* in the second sentence can refer to, it would be easy to say that the *MadeOf* relation can be extracted. However, due to the limitations of the tool, which can only coreference personal pronouns and not including *it*, it was not included in the scope of this research. Here is another instance of the same relation:

Kisha will bake a cake. She prepared the flour, butter and sugar.

In this example, there is no *MadeOf* indicator present and there is no clear indicator that what Kisha is preparing for in the second sentence refers to the cake. Through these sample sentences, it is worth noting that some concepts can have elaborations and background information provided in succeeding adjacent sentences.

4.4 Indicators

Indicators are transition words that aid in identifying explicit relations. In this study, 12 lists of indicators were created. Nine (9) are used in the templates above while the other 3 are used in the background. Table 4.7 shows the different types of indicators created.

Table 4.7: Relation Indicators

Name	**Indicators**
IsA Indicators	is a kind of, is a, is, is a type of, is an, 's a kind of, 's a, 's, 's a type of, 's an
PartOf Indicators	is a part of, is part of, has, have
MadeOf Indicators	is made of, is comprised of, form, become, becomes, forms, formed, became
OftenNear Indicators	is near, is beside
CapableOf Indicators	can, could
LocationOf Indicators	is at, is located at, can be found at, in
UsedFor Indicators	is used for, requires, required, require, with
Owns Indicators	owns, own, owned, has, have, had
Motivation Indicators	because, Because, since, Since, due, Due
Goal Indicators	because, Because, since, Since, due, Due
Cause Indicators	because, Because, since, Since, due to, Due to, for, For, as, As
Effect Indicators	as a result, As a result, Because of this, because of this, As a consequence, as a consequence, consequently, Consequently, hence, Hence, so, So, For this reason, for this reason, therefore, Therefore, thus, Thus

These sets of indicators were collated from ConceptNet (see Section 3.4), Picture Books 2 (Ang et al., 2010) (see Section 3.1.2), and sentences from the corpora. Additional indicators were from the researcher's own knowledge.

The Motivation, Goal, Cause and Effect indicators were all from Picture Books 2 (Ang et al., 2010). These groups of indicators also contain redundant indicators like *because* and *Because*. This is due to the fact that they can be found at the start or in the middle of a sentence. This is also due to the nature of the information it tries to signal which can span into sentences.

As for IsA, PartOf, PropertyOf, MadeOf, OftenNear, CapableOf, LocationOf, UsedFor and Owns indicators, the different tenses are included. It is also assume that these can only be found in the middle of sentence.

4.5 Architectural Design

The architecture of the system is mostly dependent on the open-source tool GATE (See Section 3.5). The *unstructured corpora* refers to the input children's stories. Each story must undergo the modifications mentioned in Section 4.1.1. The components shown in Figure 4.1 comprise the architectural design. Each must be included in a GATE Application Pipeline that will be run.

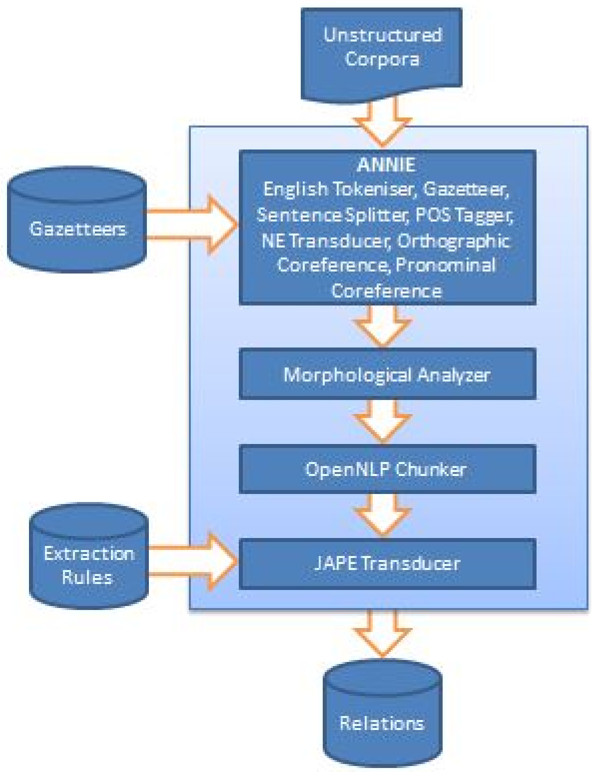

Figure 4.1: Architectural Design

Each children's story text from the *unstructured corpora* is opened in GATE by creating a GATE Document. Once the GATE documents are created, the next step is to create a GATE Corpus and add the documents. First, the GATE application resets the input of any previous annotations. This applies only if the input has already been annotated before passing through GATE. After cleaning, the input is parsed into tokens. The input was then split into sentences and each token was annotated with their respective part-of-speech tags. After that, named-entities from the defined gazetteers were annotated. Then, pronouns are matched with the named-entities they are referring to in the text.

Then, each token are processed to identify their lemmas, affixes and chunks.

Lastly, the input text is ran through a transducer. This will finally identify all the target relations and create the appropriate annotations for each.

The output of the GATE tool is an annotated version of the story that also contain the extracted target relations.

4.5.1 Resolving Story-specific Named Entities

In this implementation, a gazetteer resource was used to identify named entities in the input texts. However, the predefined lists does not cover some of the named entities in the children's story corpora. Such include characters, locations and roles. For example, in the *Winnie the Pooh* set of stories, *Pooh*, *Piglet* and *Tigger*, among others, are not included. Thus, additional lists are needed. Aside from the named entities found in stories, gazetteers are also created for indicators, world states and emotions, among others. These will be used by the transducer in its annotation patterns. A total of 20 new gazetteers were created for this study. Shown in Table 4.8 are the gazetteers created for indicators.

Table 4.8: New Gazetteers (Indicators)

File Name	**Description**
isaindicator.lst	IsA Indicators
partofindicator.lst	PartOf Indicators
madeofindicator.lst	MadeOf Indicators
oftennearindicator.lst	OftenNear Indicators
capableofindicator.lst	CapableOf Indicators
locationofindicator.lst	LocationOf Indicators
usedforindicator.lst	UsedFor Indicators
ownsindicator.lst	Owns Indicators
motivationindicator.lst	Motivation Indicators
goalindicator.lst	Goal Indicators
causeindicator.lst	Cause Indicators
effectindicator.lst	Effect Indicators

Table 4.9 shows the gazetteers created for the story characters, locations and objects. Others were added to assist in improving the extraction of target relations.

Table 4.9: New Gazetteers (Miscellaneous)

File Name	Description
storyLoc.lst	Story Locations
animal.lst	Animals
bodypart.lst	Body Parts
position.lst	Spatial markers
object.lst	Objects
character.lst	Story Characters
emotion.lst	Emotions
worldstate.lst	World States

4.5.2 Recognizing Target Relations

In order to extract the target relations, a number of JAPE phases must be created to recognise and annotate them in the input texts. A total of 23 customized JAPE phases were created for this purpose. Shown in Table 4.10 are the new JAPE phases to recognize the target relations.

Table 4.10: New JAPE Phases (Target Relations)

File Name	Description
isARelation.jape	IsA Phase
partOfRelation.jape	PartOf Phase
madeOfRelation.jape	MadeOf Phase
oftenNearRelation.jape	OftenNear Phase
capableOfRelation.jape	CapableOf Phase
locationOfRelation.jape	LocationOf Phase
usedForRelation.jape	UsedFor Phase
ownsRelation.jape	Owns Phase
effectOf.jape	EffectOf Phase
effectOfIsState.jape	EffectOfIsState Phase
eventForGoalEvent.jape	EventForGoalEvent Phase
eventForGoalState.jape	EventForGoalState Phase
happens.jape	Happens Phase
hasRoleRelation.jape	HasRole Phase
roleResponsibleForRelation.jape	RoleResponsibleFor Phase

Table 4.11 shows the other JAPE phases that are prerequisites of the target relation JAPE phases. These tag the custom tags *Event*, *Goal*, *GoalEvent*, *GoalState*, *Cause* and *Effect*. *Preprocessing1.jape* is responsible in identifying the

named-entities added for the benefit of the existing corpora.

Table 4.11: New JAPE Phases (Miscellaneous)

File Name	**Description**
cause.jape	Identify causes
effect.jape	Identify effects
event.jape	Identify events
goal.jape	Identify goals
goalEvent.jape	Identify goal events
goalState.jape	Identify goal states
preprocessing1.jape	Identify named entities based on new gazetteers

The JAPE tool follows a sequence in performing the aforementioned phases. Shown below is the sequence in tagging the custom tags and extracting the target relations:

preprocessing1
goal
goalEvent
goalState
event
cause
effect
isARelation
propertyOfRelation
madeOfRelation
oftenNearRelation
locationOfRelation
capableOfRelation
usedForRelation
ownsRelation
hasRoleRelation
partOfRelation
roleResponsibleForRelation
eventForGoalEvent
eventForGoalState
effectOf
effectOfIsState
happens

Preprocessing1.jape is the first phase ran to identify and tag all the new named-entities found in the corpora. This is followed by the tagging of *Event*, *Goal*, *GoalEvent*, *GoalState*, *Cause* and *Effect*. These are prerequisites for the *EventForGoalEvent*, *EventForGoalState*, *EffectOf*, *EffectOfIsState* and *Happens* phases. The rest of the phases were put in that sequence based on the order of their creation.

During each phase, the whole document is processed. This allows multiple relation types to be extracted from a single sentence.

4.6 Post-Processing

After running the GATE tool to annotate the stories, all post-processing were done semi-automatically. Since the output of GATE is just the annotated version of the input corpus, the annotated stories were saved as XML from the GATE program. These XML files contain all annotations like *Token* and *Sentence*, among others. The target relations were then extracted to a PBREL file.

PBREL files are created for the purpose of this study. It contains all extracted target relations from a story. Aside from the relations and their corresponding concepts, the rules used to extract them are also included for debugging purposes. An excerpt from *A wild weather day.pbrel* is shown below.

UsedFor(snacks,eating) UsedForRelation3
Owns(their,adventure) OwnsRelation3
Owns(Pierre,garden) OwnsRelation3
CapableOf(thunder,go) CapableOfRelation1
EffectOf(stopped blowing,stopped falling) EffectOf4
EffectOf(shook the clubhouse,looked out the window) EffectOf4
EffectOf(got really cold,turned into balls of ice) EffectOf4

After creating PBREL files for each story, the extracted relations were then collated into a single CPBREL (Collated PBREL) file for each big group of stories (*RAW* and *MODIFIED*). Aside from collating, redundant relations were also removed. The final CPBREL files for the *RAW* and *MODIFIED* stories were used in the testing stage.

5 Results and Analysis

This chapter discusses the overall quality of the extracted relations. It also describes the results and the methodology in evaluating them.

5.1 Methodology

This section details the strategies employed in evaluating the quality and completeness of the extracted relations.

5.1.1 Quantitative Evaluation

In order to evaluate the accuracy of the relations extracted, a gold standard was created. The gold standard consists of 3 stories selected from the *MODIFIED* corpora. Each were manually tagged with the appropriate relations. The rules in tagging the relations for the gold standard was independent from the rules followed in this research. For example, the gold standard has tagged relations that span more than 2 sentences. This was not the same for the automatically extracted relations. Lastly, only unique relations are tagged.

After completing the gold standard, the unique automatically extracted relations from the same stories were used for comparison. The extractions' precision, recall and F-measure were computed to come up with the accuracy.

Selected Stories

Each story used in developing the gold standard was selected for different reasons. Shown below are the titles of the stories and a brief description.

1. **Everybody Cries** - This was from the *Little Life Lessons* story group. It has 86 lines with a mix of simple and complex sentences. It's target audience are 4-7 year old children. It was selected for its length and complexity. Aside from this, it was determined almost all target relation types, except for *MadeOf*, can be tagged in the story.

2. **Start School** - This was from the *Topsy Tim* story group. It has 39 lines with a mix of simple and complex sentences. It's target audience are 4-7 year old children. It was selected for its complexity and target age group.

3. **Hopsalot's Garden** - This was from the *Jumpstart* story group. It has 22 lines of mostly simple sentences. It's target audience are 8-10 year old children. It was selected for its simplicity and target age group.

Metrics

In determining the accuracy and completeness of extractions, 3 metrics were used. These are precision, recall and F-measure. Precision shows the probability that an extracted relation is a true positive and expected to be extracted from the stories.

$$Precision = \frac{extractedrelations \cap expectedrelations}{extractedrelations}$$

Recall shows the probability that an expected extraction from the gold standard is actually extracted from the stories. It will show how complete the automatic extractions are.

$$Recall = \frac{extractedrelations \cap expectedrelations}{expectedrelations}$$

F-measure is the weighted average of precision and recall. In this evaluation, the balanced F-measure is computed. The best value is 1 and the worst is 0. This will determine the test's accuracy.

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

The final values were rounded off to 2 decimal places.

5.1.2 Story Evaluation

In order to validate the quality of the extracted relations, they were used to generate stories in Picture Books. First, Picture Books was examined to determine which themes can be used. Then, the target relations extracted from the *RAW* and *MODIFIED* corpora were evaluated to see whether any of them can be easily inserted into the existing Picture Books ontology. Additional relations were manually added by the researcher for the target relations which did not have valid extractions from either corpora. Then, the Picture Books databases were manually updated with a select number of valid extractions, as well as the additional ones.

Examining the Picture Books Themes

Aside from checking the valid extractions that can be used for testing, the themes of Picture Books were also examined. There are two motivations for this. First, the researcher would like to determine which among the themes can be used for testing. And secondly, he would like to map which target relation types can be tested for each theme selected.

After carefully tracing the ontology accesses and searches within the 15 different themes, five (5) were identified. Table 5.1 shows the themes selected for testing with the mapped target relation types.

Table 5.1: Mapping of Picture Books Themes with Target Relation Types

Theme ID and Lesson	**Mapped Target Relation Types**
THME0001: Take Bath	usedFor
THME0003: Be Careful	isA, capableOf, effectOf, propertyOf
THME0012: Be Honest	isA, capableOf, effectOf, propertyOf
THME0005: Be Neat	eventForGoalEvent
THME0015: Be Brave	effectOfIsState

After examination, only 7 target relation types were possible to be tested using the existing Picture Books themes.

Grouping Target Relation Types

Since only 7 target relation types can be tested by branching the existing ontology accesses and searches in the existing themes (See Section 5.1.2), another group was created. This includes the 9 remaining target relation types which were not present in any of Picture Books' existing themes. A summary of the grouping is shown in Table 5.2.

Table 5.2: Target Relation Type Groups

Group A: Used in existing themes	**Group B: Not used in existing themes**
isA, capableOf, propertyOf, effectOf, effectOfIsState, usedFor, eventForGoalEvent	locationOf, partOf, madeOf, eventForGoalState, oftenNear, happens, hasRole, roleResponsibleFor, owns

Group A Relations

For this group, 9 new lexicon entries and 12 new concepts were added into the database. The discrepancy was due to the words already existing in the lexicon but not as concepts to be used for ontology searches. Then, forty-one (41) new ontology entries were added to branch from existing ontology searches. Shown in Figure 5.1 is a sample alternative path created for this study. This is for author goal AUTH0026 of theme THME0003 (Be Careful).

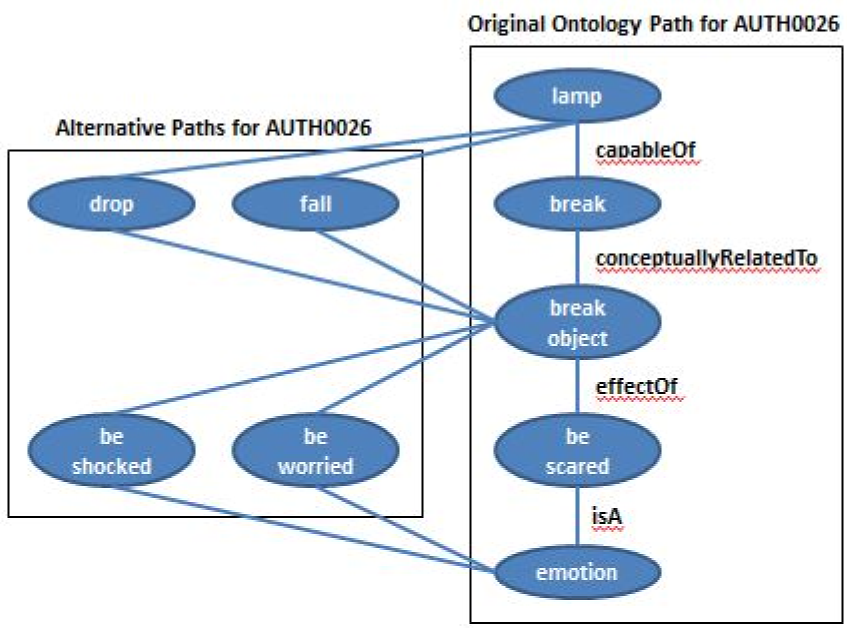

Figure 5.1: Alternative Path for AUTH0026

From the figure, there were 4 new concepts added and 8 new relations (2 *IsA*, 2 *ConceptuallyRelatedTo*, 2 *EffectOf* and 2 *CapableOf*). Though the relation *ConceptuallyRelatedTo* is not really a target relation for this study, such entries were still created to maintain the path going to the last node (*emotion*).

However, as an exceptional case in this group of relations, the *EffectOfIsState* relation was configured differently. Aside from the additional lexicon entries, concepts and ontology entries, 2 new author goals and 2 new story plot trackers were created. The new story plot trackers were then added as alternative Solutions to the theme THME0015 (Be Brave). This is all due to the difference in accessing the *EffectOfIsState* in the author goal AUTH0056. Figure 5.2 shows how the relation was accessed in the path. Instead of accessing the relation in the middle of the path, it was done at the end which was explicitly indicated in the definition of the author goal AUTH0056. This means there is no way to randomly select a new concept.

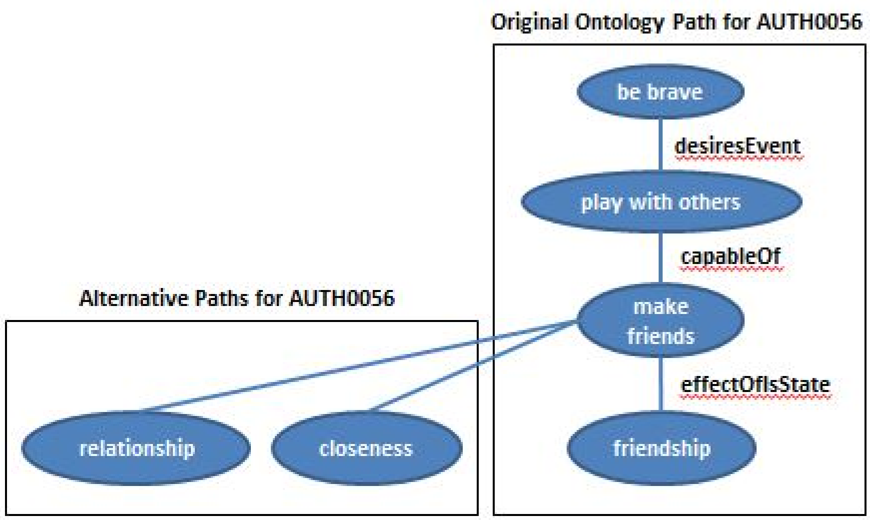

Figure 5.2: Alternative Path for AUTH0056

Appendix E shows all the additional entries in the Picture Books lexicon and ontology.

Group B Relations

Because Group B relations did not exist in the current version of Picture Books' ontology, author goals were created to show the relations in the generated stories. 16 new lexicon entries and 20 new concepts were added into the database. Then, 27 new ontology entries were added. Finally, 9 new author goals were created and inserted as additional author goals of story plot tracker SPAT0004.

For this group of relations, the theme THME0001 (Take Bath) was used. Each relation type had 2 representative sentences created and were included in the final story generated. None of the concepts and sentences were made coherent to the rest of the story. Appendix E shows all the additional entries in the Picture Books lexicon and ontology.

Selected Relations

Out of all the extracted relations from the *RAW* and *MODIFIED* corpora, only 6 relations were used and all of them are *PartOf* relations. The rest of the relations used for testing were manually created.

Group A needed extracted relations that are related to the themes selected as they are embedded in the ontology searches and accesses. But none of the extracted relations that fall under this group were coherent to any of the themes. *EffectOfIsState* was also not extracted. Thus, it was decided to just create new relations that would be coherent to the rest of the stories.

Group B, on the other hand, did not require the extracted relation to be coherent to the theme *Take Bath* because that was not the intention. But because *MadeOf* and *OftenNear* were not extracted at all and *EventForGoalState*, *Happens*, *HasRole*, *Owns* and *RoleResponsibleFor* did not have valid extractions, manual entries were also created.

5.2 Extraction Analysis

5.2.1 Accuracy and Completeness

Table 5.3: Gold Standard Evaluation Results

Stories	**Gold Standard**	**Extraction**	**Delta**	**Correct**	**P**	**R**	**F**
Overall (RAW)	663	615	-48	212	0.34	0.32	0.33
Overall (MODIFIED)	663	662	-1	241	0.36	0.36	0.36
Everybody Cries (RAW)	456	392	-64	137	0.35	0.30	0.32
Everybody Cries (MODIFIED)	456	410	-46	156	0.38	0.34	0.36
Start School (RAW)	139	150	11	47	0.31	0.34	0.33
Start School (MODIFIED)	139	173	34	54	0.31	0.39	0.35
Hopsalot's Garden (RAW)	68	73	5	28	0.38	0.41	0.40
Hopsalot's Garden (MODIFIED)	68	79	11	31	0.39	0.46	0.42

In this evaluation, the precision, recall and F-measure values were analyzed to determine whether the approach used in automatically extracting the relations yielded accurate and complete extractions.

Shown in Table 5.3 are the results after comparing the automatically extracted relations from both the raw and modified versions of the stories to the gold standard. The *Gold Standard* column shows the number of tagged relations in the gold standard for a specific story. The value is the same for the *RAW* and *MODIFIED* rows of the same story. The *Extraction* column shows the number of automatically extracted relations using this research's approach. It is followed by the *delta* which show whether there was over-extraction or under-extraction. The *Correct* column shows the number of correct automatically extracted relations after comparing with the gold standard. Columns *P*, *R* and *F* show the precision, recall and F-measure values, respectively.

Overall, the *RAW* and *MODIFIED* stories garnered 0.33 and 0.36, respectively, for their accuracy. The increase for the *MODIFIED* stories give a positive indication that the modifications done on the corpus has helped in improving the accuracy of the extractor. However, these values are still low and does not even pass the half way point. Despite cleaning the data and making sure that proper information and relations are extracted, the templates are still lacking. Appendix G shows the detailed results per relation per story.

Excess and Deficient Extractions

Focusing on the *delta* values alone, it is noticeable that there is no overall trend in the number of automatic extractions done. Only the *Everybody Cries* story had less extractions while the rest of the stories had increases.

Going into specific relation types, there are three consistent ones. *CapableOf* and *EffectOfIsState* show consistent decreases while *PropertyOf* shows consistent increases. In the case of *CapableOf*, one main reason is that the action done is not always detected. Its extraction template immediately looks for an adjacent verb as its capable action. If there are modal or other words between the doer and the action in a sentence, no relation will be extracted. Such is also the case for sentences that have complex verb phrase structures. Here is a sample sentence from *Everybody Cries*:

> *While he pretends to catch, throw, and bat, Bear bumps into Pig's swing.*

In the gold standard, there were 4 *CapableOf* relations extracted from that sentence:

CapableOf(Bear,pretend to catch)
CapableOf(Bear,pretend to throw)
CapableOf(Bear,pretend to bat)
CapableOf(Bear,bump)

The *pretends to catch, throw, and bat* were divided into 3 separate relations. But in the current templates, this is not handled.

For *EffectOfIsState*, there is a consistent 100% decrease in the number of extractions as compared to the gold standard. This was due to the fact that the template requires an adjective as an effect state but the ANNIE part-of-speech tagger is not able to tag them accurately.

Lastly, *PropertyOf* relations showed the greatest increase among the relation types. But in this case, the fault lies on the way the transducer was designed to extract this kind of relation. In its template, the transducer is supposed to relate an adjective to an adjacent noun. But unlike the rest of the relations, the transducer does not stop until all possible nouns are related to it.

Complex and Long Stories

In Table 5.3, the stories are ordered based on the length and complexity of its sentences. *Everybody Cries* has the most number of lines and the most complex sentences. *Hopsalot's Garden* has the least number of lines and the simplest sentences. If this is observed alongside their *precision*, *recall* and *F-measure* values, there is a seemingly upward trend. The length of the story and complexity of the sentences are indirectly proportional to the accuracy of extraction. As the complexity of the sentences (with the introduction of more clauses) and the number of lines increases (longer stories), the extractor suffers from lower accuracy.

Correct and Complete Extractions

Overall, the extractor yielded fairly low *precision*, *recall* and *F-measure* values. This shows that the templates used, despite their generalized nature to accommodate a number of patterns, were not able to extract an acceptable number of valid relations. And because of this, the templates were not successful in extracting all the expected valid relations from the stories.

For those relation types which were able to extract more than the expected, the way the transducer was designed to extract them did not help in improving the quality of relations. It introduced more nuisance relations than expected. Another contributing factor would be their loose and highly generalized templates. As for those that had lower extractions than the gold standard, their templates may have been a bit stricter but it also caused the extractor to miss a lot of valid extractions.

And for most relation types, another reason would be their limitation to be extracted from a single sentence only. The gold standard had a lot of inferred and implied tagged relations that span even more than 2 sentences. These would be impossible to extract using the current approach. Also, almost all gold standard relations were tagged not using much of the defined indicators in this study.

However, there was a bright spot amongst the relation types. The *Owns* relation type was consistent in getting *precision* and *recall* values ranging from 0.48 to even 0.77. It only got a *precision* of 0.48 in *Everybody Cries* while in the other 2 stories, values were always above 0.62. Though in a broader sense, the scores are still low and not acceptable, it is worth noting how this relation type consistently bested the other relations in terms of extractions.

First, *Owns* relations are easy to distinguish in a sentence. When a possessive noun or pronoun is present, it is highly likely that the noun adjacent to it would constitute your second concept. Also, this relation type, unlike most, is almost always found within a sentence. And if it is within a span of text, there are only a few indicators possible and the ways it can be expressed are limited. Lastly, its templates are simpler and straightforward. There was not a lot of generalization done because there are not a lot of ways that an *Owns* relation is manifested in a sentence. However, it cannot handle possessions (second concept) that have more than 1 word specially when adjectives are used. For example, the relation *Owns(Bear,first game)* was not extracted because the template was immediately looking for a noun. The presence of the adjective *first* hindered this extraction.

Low Correct Extractions

In this section, only the *Everybody Cries* story is considered since it was the only story that have almost all relation types tagged in the gold standard.

Though the extractor was able to identify relations for almost all types except for *MadeOf*, *EffectOfIsState* and *OftenNear*, there were some relation types that have very low correct extraction while some did not have any correct extractions at all. First in the list would be the *IsA* relation. In the gold standard, all relations of this type were implied and spans across sentences. The use of indicators were also not apparent. Here is an example:

> *Puppy likes to have a good breakfast before school.*
> *This morning he has no time for cereal, toast, or juice.*

In the gold standard, it was able to identify the following *IsA* relations:

> *IsA(cereal,breakfast)*
> *IsA(juice,breakfast)*
> *IsA(toast,breakfast)*

The current *IsA* templates cannot handle this since it is limited to a sentence and it highly relies on indicators.

Next, *PartOf* relations had a very low turnout of correct relations because of its similarity to *Owns*. One of its template uses possessive nouns or pronouns to signal its presence. However, the extractor is only able to differentiate it from an *Owns* extraction when the second concept or the possession is a body part. In cases like *car's wheel* and *building's glass windows*, it wasn't able to identify the *PartOf* relation. Instead, these are incorrectly tagged as *Owns*.

EventForGoalEvent is another case of confusion. In most cases, these types of relations are identified by the extractor as *EffectOf* because of the similarity in structure when apparent in a text. Because desires are not always explicitly shown, the extractor had a difficult time differentiating this relation from *EffectOf* which can accept just events as cause and effect if there are no indicators used. Another reason why this relation had a low turnout was that despite being able to extract from 2 sentences, most of the gold standard relations were extracted beyond 2 sentences. Here is an excerpt from the story:

> *They will all share their lunches with Puppy.*
> *Puppy's friends pass him bits from their lunches.*
> *Hippo shares his sandwich with Puppy.*
> *Bear gives him some crackers.*
> *Mouse shares her pudding.*
> *Cat gives him part of her apple.*

The gold standard was able to identify the following relations:

> *EventForGoalEvent(give cracker,share lunch)*
> *EventForGoalEvent(give part of apple,share lunch)*
> *EventForGoalEvent(pass bits,share lunch)*
> *EventForGoalEvent(share pudding,share lunch)*
> *EventForGoalEvent(share sandwich,share lunch)*

In this specific example, the relations were present even if they were 4 sentences apart. Clearly, the extractor was not able to handle that. Aside from that, there was no clear indication in the first sentence that *share lunch* was the desire of Puppy's friends.

The *UsedFor*, *Happens* and *HasRole* relations also had zero correct extractions. *UsedFor* was not able to correctly extract the expected relations because its templates were always expecting an indicator. But in the story, all *UsedFor* instances did not use indicators. Here is an example:

> *She is writing on the blackboard and does not turn around to look at Puppy.*

The gold standard was able to identify the relation *UsedFor(blackboard,write)* but this was not present in the actual extractions. The existing templates should have been able to identify this if no indicators were expected.

As for the *Happens* relation, the extractor always extracted a relation when the *today* time indicator was used in a sentence. But in the gold standard, *today* was not considered a valid time indicator as it is vague and does not really give a valid time within a day.

HasRole is one of those relations that is quite rare and hard to find in a story. In the gold standard, there was only 1 *HasRole* found and it is *HasRole(Miss Hen,teacher)*. This was not extracted using the extractor. First, the templates are looking for indicators and certain structures for it to identify the relation. But in this case, the correct relation was identified by inferring from the actions done by *Miss Hen* in the story. The role was also not mentioned in the story. The gold standard used the following actions in determining the *teacher* role for *Miss Hen*: *ask the class*, *collect assignment*, *start class* and *write on the blackboard*. These are specific actions that are done by a teacher which is why the said role was related to the character.

Lastly, the *OftenNear* relation did not have any correct extractions because of insufficient indicators. The only *OftenNear* relation identified in the gold standard was *OftenNear(house,sidewalk)*. This was identified using the word *along* which was not included as one of the *OftenNear* indicators. Had it been added in the first place, the said relation would have existed as part of the extractions.

5.2.2 Uniqueness and Redundancies

In this evaluation, the number of unique extractions were analyzed to determine whether the modifications created more relations valid for extraction. Also, duplicate or redundant extractions are also analyzed to see whether they were reduced after modification.

There was a total of 13,871 unique relations extracted from the *RAW* and *MODIFIED* corpora. Tables 5.5 and 5.4 show the breakdown of numbers for each group of stories in each corpora.

Table 5.4: Raw Corpus: Number of Unique and Redundant Relations Extracted

Story Group	**Unique**	**Redundant**
Jumpstart	644	194
Winnie the Pooh	394	94
Topsy Tim	684	215
Little Life Lessons	4977	2017
TOTAL	**6699**	**2520**

Table 5.5: Modified Corpus: Number of Unique and Redundant Relations Extracted

Story Group	**Unique**	**Redundant**
Jumpstar t	729	206
Winnie the Pooh	436	137
Topsy Tim	747	341
Little Life Lessons	5260	2621
TOTAL	**7172**	**3305**

From these numbers, it is evident that the *Modified* corpus had 473 more relations extracted than the *Raw* corpus. This suggests that the modifications done introduced new concepts, thus allowing the application to extract more unique relations. These modifications may have also introduced more redundant relations as the number increased by 785. This was even more than the increase in unique relations. Here are sample passages from *CJ and the Mysterious Map*:

"And how do you suggest we get up there?" asked Edison.
"What are you doing?" asked Edison.

These are non-adjacent sentences from the story. After running this in the extraction tool, they yielded zero extractions. Then they were modified to this:

Edison asked CJ some suggestions.
Edison asked what CJ is doing.

Now, each sentence was able to produce 2 new *CapableOf* extractions. However, both of them are *CapableOf(Edison,ask)*. And after examining the modified story, there were more sentences of this structure that yielded the same relation.

In terms of relations, Table 5.6 shows the breakdown of extracted relations per corpora. The delta after modification is also shown.

Table 5.6: Number of Relations Extracted per Corpus

Relations	**Raw**	**Modified**	**Delta**
IsA	191	228	37
PropertyOf	2700	2857	157
PartOf	71	84	13
MadeOf	0	0	0
EventForGoalEvent	481	580	99
EventForGoalState	31	37	6
EffectOf	1258	1196	-62
EffectOfIsState	0	0	0
CapableOf	980	1193	213
OftenNear	0	0	0
LocationOf	136	137	1
UsedFor	228	232	4
Happens	38	34	-4
HasRole	11	14	3
RoleResponsibleFor	11	13	2
Owns	562	565	3

In a relational level, almost all relations experienced an increase in the number of extractions except for *EffectOf* and *Happens*. Again, although the delta is not significant in number, it suggests that the modifications avoided the extraction of possible relations. Here is an extracted relation from the *Raw* corpus that is not included as an extracted relation in the *Modified* corpus:

EffectOf(said,not forgive and forget)

It was extracted from this passage from *Forgive and Forget* of the *Winnie the Pooh* story group:

"Rabbit, I think Tigger is very sorry," Pooh said. "Will you not forgive and forget?"

This specific sentence was modified to this:

Pooh thinks Tigger is very sorry. Pooh asks Rabbit if he can forgive and forget.

In this example, the verb *said* was removed after modification, thus avoiding the possibility of another relation being extracted. It seems that modifications, such as this, improved the result for the *Modified* corpus because the relation is not really a valid and logical extraction. However, the possibility that valid extractions may also be omitted in the process cannot be discounted.

5.2.3 Zero Extraction

From the same table, it is important to note that the *EffectOfIsState*, *MadeOf* and *OftenNear* relations had 0 extractions from either corpora. This was caused by their high dependence on indicators, incorrect part-of-speech tags and limitation on the number of sentences it can extract from.

5.2.4 Quality

In terms of extraction quality, there are some relations which may have been incorrectly extracted or which may not be extracted at all because of incorrect tags. For instance, this relation was extracted from the *Jumpstart* story group:

EffectOf(went on full speed,shocked)

The extraction seems valid from first glance, but looking at its original passage may suggest otherwise. Shown below is the modified passage from *Just in time*. Instead of *EffectOf*, a better extraction could have been *EffectOfIsState* since *being shocked* is an implied state.

Frankie went on full speed.
Hopsalot was shocked.

Other incorrect extractions were due to a different part-of-speech information tagged to the concept and the lack of additional semantic information.

Lastly, it is important to note that for event relations like *EffectOf* and *Event-ForGoalEvent*, the extracted relations seem to be longer and more specific because the extractor uses whole phrases as concepts. This may be different from the concepts of Picture Books that are more generalized. Here are some example extractions:

EffectOf(looked at the map,checked the wind)
EffectOf(pours something into the volcano,stopped him)
EventForGoalEvent(called everyone,go to the ship)

For a more detailed look in the extraction results, kindly refer to Appendix D.

5.3 Story Analysis

Overall, the new relations were used accordingly. They were randomly selected and used interchangeably with the original relations. But the sentences generated were fairly acceptable. There are cases wherein the sentence is grammatically incorrect. For example, here is an original story generated for the *Be Careful* theme:

Edward the elephant learns to be careful.

```
[1]  The morning was sunny.
[2]  Edward the elephant was in the dining room.
[3]  He played near breakable glass of water.
[4]  Daddy Sam told Edward to be careful.
[5]  Edward continued to play glass of water near.
[6]  He broke it.
[7]  Edward was scared.
[8]  He hid away from Daddy Sam.
[9]  Daddy Sam saw that glass of water was broken.
[10] Daddy Sam called Edward.
[11] Daddy Sam told Edward that he should have obeyed.
[12] He felt sorry.
[13] Daddy Sam cleaned up glass of water.
[14] Edward helped Daddy Sam to clean up.
[15] Daddy Sam reminded Edward to be careful.
[16] Being careful is important.
[17] From that day onwards, Edward always was careful.
```

This used the original relations of Picture Books. The highlighted lines show the sentences where the changes should happen. Shown below is another story using the same characters and objects as the story above. This time, the highlighted sentences are using different concepts based on the new additions to the lexicon and ontology.

Edward the elephant learns to be careful.

```
[1]  The afternoon was sunny.
[2]  Edward the elephant was in the dining room.
[3]  He played near breakable glass of water.
[4]  Mommy Edna told Edward to be careful.
[5]  Edward continued to play glass of water near.
[6]  Edward fell it.
[7]  Edward was worried.
[8]  He hid away from Mommy Edna.
[9]  Mommy Edna saw that glass of water was cracked.
[10] Mommy Edna called Edward.
[11] Mommy Edna told Edward that he should have obeyed.
[12] Edward felt sorry.
[13] Mommy Edna cleaned up glass of water.
[14] Edward helped Mommy Edna to clean up.
[15] She reminded Edward to be careful.
[16] Being careful is important.
[17] From that day onwards, Edward always was careful.
```

In line 6, instead of the usual action *broke*, the story now uses *fell*. However, the new sentence should have been "It fell" or "The glass of water fell." Since the relation used here is *CapableOf(lamp,fall)*, it also logical to note that the *lamp* can be the agent of the action instead of always being the character.

Another new story worth pointing out is this:

Edward the elephant learns to be honest.

```
[1]  The morning was warm.
[2]  Edward the elephant was in the dining room.
[3]  He played near breakable glass of water.
[4]  Edward fell glass of water.
[5]  He was worried.
[6]  Mommy Edna saw that glass of water was smashed.
[7]  Edward told Mommy Edna that Porky the pig broke glass of water.
[8]  He was sad.
```

```
[9]  Porky cried.
[10] Edward felt guilty.
[11] Edward told Mommy Edna that he dropped glass of water.
[12] Mommy Edna told Edward that he should have been honest.
[13] He apologized to Mommy Edna.
[14] Edward apologized to Porky.
[15] Mommy Edna told Edward to be honest.
[16] Mommy Edna told Edward that being honest is good.
[17] From that day onwards, Edward always was honest.
```

In this story, highlighted are the sentences using the *CapableOf* relations with *lamp* as the parent concept, *CapableOf(lamp,fall)* and *CapableOf(lamp,drop)*. Though it is valid to use both, having them in just one story creates inconsistencies. In line 4, it was already mentioned that the glass of water fell. And since the word *fell* was used, there is an implication of an accident which creates a different dimension to the story. Now in line 11, though Edward is already admitting his fault, he suddenly said *dropped* instead of the initial *fell*. This might create confusion as dropping something makes the act intentional whereas when it fell, Edward may or may not have caused the action. When the act *fell* was used in line 4, it expected that the same act is confessed in line 11.

Lastly, there are a number of instances where incorrect relations are picked up by Picture Books in doing ontology accesses and searches. As a result, incorrect sentences are generated. For example, in the story *Roy the chicken learns to take bath.* shown below, lines 9 and 10 are produced using the same relation, *OftenNear*. They also have the same parent concept which is *school*. Line 9 has a correct child concept but Line 10 does not. Instead of either *clinic*, *mall* or *market*, it used *generic*. After examining the ontology, *generic* is the child concept when the relation is *PropertyOf*. This shouldn't be the case.

The reason why this happens is because Picture Books searches its ontology not by the name of the relation but by its category. In the ontology, both *OftenNear* and *PropertyOf* are under the category *spatial*. This causes Picture Books to not only randomize among *OftenNear* relations but also including the *PropertyOf* relation.

```
[1]  Toys were the bedroom.
[2]  Playing is the playground.
[3]  An oak had trunk.
[4]  A person had a toe.
[5]  A book had paper.
[6]  It had ink.
```

```
[7]  Playing is dirty.
[8]  It was healthy.
[9]  The school was the market.
[10] It was generic.
[11] Eating dinner is the evening.
[12] Going to the school is the morning.
[13] A fireman was rescuing.
[14] A librarian was organizing.
[15] Daddy Sam had a ball.
[16] He had the tricycle.
[17] From then on, Roy always took the bath.
```

Another example is with the same story but lines 11 and 12 are changed to the following:

```
[11] Eating dinner is the evening.
[12] Going to the school is saying goodbye.
```

These two sentences are using the *Happens* relation. But in line 12, the relation *FirstSubeventOf* was used because they are both *event* relations. This happens for ontology accesses that only have 1 argument.

Kindly refer to Appendix F for all the stories generated in this phase.

6 Conclusion and Recommendations

This chapter discusses the conclusion of this research and provides recommendations and suggestions for future researches.

6.1 Conclusion

Based on the results obtained through the evaluation of the extractor, it was proven possible to extract new semantic relations from children's stories and feed them into Picture Books' ontology. However, the extractor was found to be inaccurate in doing so. Overall, it only got 0.36 as its precision, recall and F-measure scores for 3 MODIFIED stories. It even got lower scores for the RAW versions. It got 0.34, 0.32 and 0.33 for its precision, recall and F-measure, respectively. Therefore, the automatically extracted relations were mostly incorrect and the extractor was not able to extract all expected relations in a given text.

New relations were used accordingly and interchangeably, and new sentences were inserted. However, due to the limited themes currently present in Picture Books, not all extracted relations was used. There were also cases wherein all extracted relations won't be valid for Picture Books' use because the story where they came from are completely different from the existing themes. Another issue was Picture Books' way of accessing its ontology. Instead of using the relation names as reference, the relation category is used. Since each category has more than 1 relation associated, incorrect searches may still arise. Thus rendering the attempt to improve Picture Books' conceptual knowledge less remarkable. Lastly, it is not enough to just add new concepts and relations in the current ontology. Additional steps must be taken depending on how the ontology path you are trying to branch is accessed.

As for the extracted relations, their quality was greatly affected by the following:

6.1.1 Stories

Each story in the corpora has different characteristics. Some are lengthy while some are short. Some use a lot of complex sentence structures while others kept it simple. It all depends on the age group of the audience they are trying to reach. After evaluation, it is conclusive that as the sentence structures become more complex and the length of the story increases, the extractions get less accurate.

It exposes a limitation on the templates used as they can only successfully handle simpler sentences and simpler manifestations of a relation in a text.

6.1.2 Part-of-speech Tags

The quality and accuracy of the part-of-speech tags supplied by GATE greatly affected the relations extracted. Because most of the extraction rules/templates mainly use part-of-speech tags in their annotation patterns, a slight mistake may cause the relation to have incorrect concepts or to have it not extracted at all. Here is a sample sentence to illustrate this scenario:

> *Tigger takes a bath because he wants to be clean.*

One relation that can be extracted from this would be *EventForGoalState(takes a bath,be clean)*. However, after numerous attempts in the application, that relation will not be extracted because *clean* is tagged as a verb. The researcher's rule for *EventForGoalState* requires the child concept to be an adjective for it to be called a desired state.

6.1.3 Extraction rules

Because the current set of extraction rules are generalized based on ConceptNet sentence patterns and the sentences present in the current corpora, there is a perceived limit in the capabilities of the extractor. And in the attempt to cover all sentence patterns with the least number of rules, exceptional cases may not be covered. Also, these rules cannot handle implied and inferred relations. If there are any implied or inferred ones in the text, the rules won't recognize them unless they were explicitly indicated after modification. Additionally, these rules do not have enough semantic information for most words. When a different sense of a word is used in a sentence, there is no way for the extractor to recognise it. It will rely solely on the part-of-speech and named-entity tags to extract a relation. This deficiency in semantic information also causes an incorrect relation to be tagged to a concept pair. For example, *PartOf* relations can be incorrectly tagged as an *Owns* relation because of the similarity in templates used. Lastly, the templates were also limited to 1 or 2 sentences only. Most of the relations encountered in the gold standard were identified from sentence more than 2 sentences apart.

6.1.4 Indicators

The prevalent use of indicators in most of the extraction templates posed a limitation on the number and quality of extractions done. First, in most cases, indicators are not always used because of their formality. This also assumes that the concepts constituting a relation is within a sentence. If not, it is assumed that the second concept is in the next sentence, the subject pronoun referring to the first concept, and the whole thing signalled by an indicator.

Secondly, most relations identified in the gold standard were inferred or implied. Taking a look into the *HasRole* relation again, the only instance did not have both concepts present in the story. In the relation *HasRole(Miss Hen,teacher)*, the word *teacher* was not found in the text. It was inferred as a role through the actions done by the character. Lastly, existing indicators are not enough. There's still a number of indicators not included in this study. This caused the only expected *OftenNear* relation in *Everybody Cries* to not be extracted. It was signalled by the word *along* which is not part of the *OftenNear* indicators.

6.2 Recommendations

Overall, the relation extractor was able to produce good enough relations to be used by any story generation system. The following are recommendations for future improvement:

- Improve the extraction rules. Incorporate as many patterns as possible. If allowable, run a big corpora through a machine learning tool that will learn all possible sentence patterns for each relation type.
- Allow inferencing between relations as they are extracted. This will improve the compactness of the ontology.
- Focus more on extracting event relations since they are not usually explicitly indicated in a span of text. This also constitutes the bulk of a story. Building an accurate cause-effect chain of events would be very beneficial for most creative text generation systems.
- Look for a language resource that can supply accurate semantic information.
- Since a gold standard was already utilized in evaluating this research, it would be beneficial to look for experts that can create different sets of gold standards for the different relations to improve the evaluation of the results.

These sets could then be combined to form a comprehensive gold standard. Aside from this, have experts that can consistently evaluate the generated story after adding new relations.

And because part of the evaluation involved the use of Picture Books in generating stories, some issues and limitations were encountered. The following are recommendations for Picture Books' future improvement:

- Since it is now possible to branch out in ontology searches, it would be advisable to keep track of previously chosen concepts that is expected to be used throughout the story. This will avoid conflicting details and make the output stories more coherent.
- To be able to use all extracted relations, devise a way to automatically add new themes in Picture Books based on the themes found in the corpora.
- Instead of searching the ontology by category, use the relation names for more specific and accurate ontology access results.
- Integrate VerbNet to make the character goals dynamic in creating sentences. This will provide information on what arguments are needed in a sentence for a certain verb.

A Resource Persons

Ms. Ethel Ong
Adviser
College of Computer Studies
De La Salle University-Manila
ethel.ong@delasalle.ph

B Personal Vitae

Mr. Briane Paul V. Samson
11 Gen. Arellano St., West Rembo, Makati City
0917-8319462
briane.samson@yahoo.com

C Children's Stories

Table C.1 shows the different stories used in the corpus.

Table C.1: Children's Stories

Story Code	**Title**	**Story Group**
S1	A wild weather day	Jumpstart
S2	CJ and the mysterious map	Jumpstart
S3	Eleanor's enormous ears	Jumpstart
S4	Hopsalot's garden	Jumpstart
S5	Just in time	Jumpstart
S6	Lost and found in Jumpstart town	Jumpstart
S7	Rain, rain, go away	Jumpstart
S8	Everyone is special	Winnie the Pooh
S9	Forgive and Forget	Winnie the Pooh
S10	Go to the park	Topsy Tim
S11	Have new bikes	Topsy Tim
S12	In the gym	Topsy Tim
S13	Learn to swim	Topsy Tim
S14	Start school	Topsy Tim
S15	Everybody Cries	Little Life Lessons
S16	Friends Share	Little Life Lessons
S17	Game Day	Little Life Lessons
S18	Good Manners	Little Life Lessons
S19	Helping Hands	Little Life Lessons
S20	Helping Out	Little Life Lessons
S21	Imagination	Little Life Lessons
S22	Learning Something New	Little Life Lessons
S23	Litterbug Bear	Little Life Lessons
S24	Pup Tent Pals	Little Life Lessons
S25	Silly Stunts	Little Life Lessons
S26	The Broken Flowerpot	Little Life Lessons
S27	The Lost Shoes	Little Life Lessons
S28	The New Kid	Little Life Lessons
S29	The Secret Fort	Little Life Lessons
S30	Treasure Hunt	Little Life Lessons

D Detailed Extraction Results

The following tables show the detailed results of the relation extraction.

Table D.1: Detailed Overall Result

	Raw		**Modified**	
Story Codes	**Unique**	**Redundant**	**Unique**	**Redundant**
S1	92	10	111	35
S2	267	61	218	36
S3	51	14	48	9
S4	75	18	68	16
S5	117	52	88	34
S6	58	23	54	41
S7	69	28	57	23
S8	232	66	213	50
S9	204	71	181	44
S10	142	58	125	38
S11	143	54	113	30
S12	200	93	191	56
S13	124	58	118	43
S14	138	78	137	48
S15	404	120	384	95
S16	305	98	258	65
S17	329	102	317	77
S18	381	162	356	101
S19	285	195	257	141
S20	350	154	399	182
S21	324	124	297	93
S22	335	268	313	190
S23	291	120	304	106
S24	326	149	320	126
S25	301	186	274	120
S26	299	132	278	122
S27	314	255	282	146
S28	298	221	261	174
S29	405	213	386	181
S30	313	122	291	98
TOTAL	7172	3305	6699	2520

Table D.2: Raw Corpus: Detailed Overall Result per Relation (Part 1)

Story Code	IsA	PropertyOf	PartOf	MadeOf	CapableOf	OftenNear	LocationOf
S1	1	36	1	0	31	0	3
S2	6	86	2	0	43	0	5
S3	4	13	3	0	12	0	0
S4	2	31	0	0	11	0	0
S5	3	24	1	0	26	0	2
S6	2	21	2	0	8	0	4
S7	4	17	1	0	14	0	2
S8	5	71	0	0	44	0	4
S9	2	71	2	0	35	0	2
S10	3	65	1	0	12	0	4
S11	1	42	0	0	21	0	0
S12	4	60	3	0	49	0	6
S13	3	54	3	0	18	0	2
S14	2	82	0	0	15	0	1
S15	13	128	9	0	76	0	7
S16	11	81	1	0	40	0	1
S17	3	133	2	0	49	0	4
S18	13	151	4	0	47	0	2
S19	3	86	1	0	27	0	0
S20	13	163	9	0	62	0	1
S21	11	109	3	0	35	0	10
S22	9	142	0	0	30	0	9
S23	6	109	0	0	50	0	6
S24	7	159	0	0	29	0	9
S25	8	110	3	0	36	0	6
S26	16	100	3	0	35	0	6
S27	6	100	2	0	36	0	8
S28	5	105	5	0	32	0	7
S29	16	197	6	0	35	0	16
S30	9	154	4	0	22	0	9
TOTAL	191	2700	71	0	980	0	136

Table D.3: Raw Corpus: Detailed Overall Result per Relation (Part 2)

Story Code	**EventFor-GoalEvent**	**EventFor-GoalState**	**EffectOf**	**EffectOf-IsState**
S1	3	0	26	0
S2	12	0	36	0
S3	1	0	9	0
S4	1	0	9	0
S5	3	0	17	0
S6	1	0	7	0
S7	1	0	15	0
S8	14	3	44	0
S9	15	1	35	0
S10	5	0	18	0
S11	8	0	19	0
S12	11	0	28	0
S13	8	1	17	0
S14	9	0	13	0
S15	23	3	64	0
S16	20	1	68	0
S17	26	0	66	0
S18	27	4	55	0
S19	33	2	56	0
S20	36	0	72	0
S21	26	5	61	0
S22	32	1	59	0
S23	16	5	74	0
S24	16	2	57	0
S25	27	0	45	0
S26	20	0	66	0
S27	23	1	62	0
S28	29	1	56	0
S29	17	0	49	0
S30	18	1	55	0
TOTAL	481	31	1258	0

Table D.4: Raw Corpus: Detailed Overall Result per Relation (Part 3)

Story Code	UsedFor	Happens	HasRole	RoleResponsibleFor	Owns
S1	6	0	0	0	3
S2	7	0	1	1	19
S3	3	0	0	0	3
S4	1	1	0	0	12
S5	1	0	1	0	10
S6	4	0	0	0	5
S7	0	0	0	0	3
S8	8	0	0	0	20
S9	2	0	0	0	16
S10	5	0	0	3	9
S11	3	0	0	0	19
S12	14	0	0	0	16
S13	3	0	1	0	8
S14	1	0	1	0	13
S15	5	10	1	3	42
S16	11	7	0	0	17
S17	14	1	0	0	19
S18	15	2	1	0	35
S19	17	4	1	1	26
S20	16	6	0	0	21
S21	12	0	1	2	22
S22	5	0	0	0	26
S23	6	1	0	0	31
S24	12	0	0	0	29
S25	9	0	0	0	30
S26	3	3	3	0	23
S27	19	0	0	0	25
S28	8	1	0	0	12
S29	8	2	0	0	40
S30	10	0	0	1	8
TOTAL	228	38	11	11	562

Table D.5: Modified Corpus: Detailed Overall Result per Relation (Part 1)

Story Code	**IsA**	**PropertyOf**	**PartOf**	**MadeOf**	**CapableOf**	**OftenNear**	**LocationOf**
S1	1	36	1	0	27	0	2
S2	9	99	2	0	64	0	5
S3	4	14	3	0	14	0	0
S4	2	33	0	0	17	0	0
S5	2	29	1	0	41	0	3
S6	2	21	2	0	10	0	4
S7	4	17	1	0	20	0	2
S8	6	77	3	0	58	0	4
S9	3	74	3	0	50	0	2
S10	3	66	1	0	24	0	2
S11	4	40	0	0	36	0	0
S12	4	58	3	0	48	0	5
S13	3	53	4	0	21	0	2
S14	2	80	0	0	18	0	1
S15	11	144	10	0	82	0	8
S16	12	101	1	0	49	0	2
S17	7	141	2	0	57	0	4
S18	12	152	7	0	68	0	1
S19	7	95	2	0	37	0	0
S20	15	141	9	0	61	0	1
S21	13	125	3	0	41	0	10
S22	12	155	0	0	33	0	11
S23	8	111	0	0	45	0	7
S24	8	164	0	0	27	0	9
S25	10	120	5	0	39	0	6
S26	17	117	4	0	42	0	6
S27	9	105	3	0	55	0	8
S28	8	118	5	0	47	0	6
S29	19	212	5	0	35	0	16
S30	11	159	4	0	27	0	10
TOTAL	228	2857	84	0	1193	0	137

Table D.6: Modified Corpus: Detailed Overall Result per Relation (Part 2)

Story Code	**EventFor-GoalEvent**	**EventFor-GoalState**	**EffectOf**	**EffectOf-IsState**
S1	3	0	14	0
S2	13	2	46	0
S3	1	0	9	0
S4	4	0	7	0
S5	8	0	22	0
S6	1	0	9	0
S7	5	0	17	0
S8	14	3	39	0
S9	16	1	35	0
S10	9	0	20	0
S11	20	0	25	0
S12	17	0	35	0
S13	13	1	15	0
S14	11	0	15	0
S15	24	3	61	0
S16	26	1	70	0
S17	31	2	47	0
S18	32	4	49	0
S19	35	2	56	0
S20	35	0	50	0
S21	28	7	55	0
S22	37	0	52	0
S23	17	4	60	0
S24	19	2	58	0
S25	31	2	46	0
S26	22	0	61	0
S27	32	1	60	0
S28	31	1	61	0
S29	20	0	46	0
S30	25	1	56	0
TOTAL	580	37	1196	0

Table D.7: Modified Corpus: Detailed Overall Result per Relation (Part 3)

Story Code	**UsedFor**	**Happens**	**HasRole**	**RoleResponsibleFor**	**Owns**
S1	4	0	0	0	3
S2	6	0	1	1	19
S3	3	0	0	0	3
S4	1	0	0	0	11
S5	1	0	1	1	8
S6	4	1	0	0	4
S7	0	0	0	0	3
S8	8	0	0	0	20
S9	2	0	0	0	18
S10	7	0	0	3	7
S11	3	0	0	0	15
S12	15	0	0	0	15
S13	4	0	1	0	7
S14	1	0	1	0	9
S15	3	9	1	3	44
S16	15	8	0	0	20
S17	14	1	0	0	23
S18	15	2	2	0	37
S19	17	3	1	1	29
S20	10	4	0	0	24
S21	12	0	3	2	25
S22	7	0	0	0	28
S23	6	1	0	0	32
S24	15	0	0	0	24
S25	12	0	0	0	30
S26	3	2	3	0	22
S27	17	0	0	0	24
S28	8	1	0	0	12
S29	8	2	0	0	42
S30	11	0	0	2	7
TOTAL	232	34	14	13	565

E New Ontology Entries

The following tables show the actual ontology entries added during testing.

E.1 Group A Relations

Table E.1: New Lexicon Entries (Group A)

Word	**WORD ID**	**ONTO ID**
fall	WORD0440	ONTO0242
drop	WORD0441	ONTO0243
cracked	WORD0442	ONTO0244
smashed	WORD0443	ONTO0245
knick knack	WORD0444	ONTO0246
rejoice	WORD0445	ONTO0247
revel	WORD0446	ONTO0248
relationship	WORD0447	ONTO0249
closeness	WORD0448	ONTO0250
be worried*	WORD0083 WORD0374	ONTO0251
be shocked*	WORD0083 WORD0414	ONTO0252
game*	WORD0344	ONTO0253

Note: *ONLY an entry in the concept table was added. Word is already existing in the lexicon.

Table E.2: New Ontology Entries (Group A) (Part 1)

OntoID	**SemanticRelation**	**Element2**	**Category**
ONTO0074	capableOf	ONTO0242	action
ONTO0074	capableOf	ONTO0243	action
ONTO0075	capableOf	ONTO0242	action
ONTO0075	capableOf	ONTO0243	action
ONTO0242	conceptuallyRelatedTo	ONTO0078	generic
ONTO0242	isA	ONTO0077	things
ONTO0243	conceptuallyRelatedTo	ONTO0078	generic
ONTO0243	isA	ONTO0077	things
ONTO0244	isA	ONTO0098	things
ONTO0245	isA	ONTO0098	things
ONTO0246	exampleOf	ONTO0066	things
ONTO0246	exampleOf	ONTO0067	things
ONTO0246	exampleOf	ONTO0068	things
ONTO0247	eventRequiresObject	ONTO0002	event
ONTO0247	eventRequiresObject	ONTO0246	event
ONTO0247	eventRequiresObject	ONTO0253	event
ONTO0248	eventRequiresObject	ONTO0002	event
ONTO0248	eventRequiresObject	ONTO0246	event
ONTO0248	eventRequiresObject	ONTO0253	event
ONTO0249	lastSubeventOf	ONTO0151	event
ONTO0250	lastSubeventOf	ONTO0151	event

Table E.3: New Ontology Entries (Group A) (Part 2)

OntoID	SemanticRelation	Element2	Category
ONTO0251	desiresEvent	ONTO0190	goal
ONTO0251	isA	ONTO0156	things
ONTO0252	desiresEvent	ONTO0190	goal
ONTO0252	isA	ONTO0156	things
ONTO0253	exampleOf	ONTO0069	things
ONTO0253	exampleOf	ONTO0070	things
ONTO0253	exampleOf	ONTO0071	things
ONTO0253	exampleOf	ONTO0220	things
ONTO0191	propertyOf	ONTO0244	things
ONTO0191	propertyOf	ONTO0245	things
ONTO0001	eventRequiresObject	ONTO0246	event
ONTO0001	eventRequiresObject	ONTO0253	event
ONTO0080	eventForGoalEvent	ONTO0247	event
ONTO0080	eventForGoalEvent	ONTO0248	event
ONTO0149	effectOfIsState	ONTO0249	action
ONTO0149	effectOfIsState	ONTO0250	action
ONTO0210	conceptuallyRelatedTo	ONTO0249	generic
ONTO0210	conceptuallyRelatedTo	ONTO0250	generic
ONTO0078	effectOf	ONTO0251	action
ONTO0078	effectOf	ONTO0252	action

Table E.4: New Author Goal Entries (Group A)

GoalID	AUTH0087
Name	Result of lesson is friendship
Goal	CGOL0013(Target:%ontoGoal(%lesson%,WORD0447)%);
Consequence	CGOL0044(Target:WORD0332);
GoalID	AUTH0088
Name	Result of lesson is friendship
Goal	CGOL0013(Target:%ontoGoal(%lesson%,WORD0448)%);
Consequence	CGOL0044(Target:WORD0332);

Table E.5: New Story Plot Tracker Entries (Group A)

PlotID	SPAT0052	SPAT0053
Name	Inform Lesson	Inform Lesson
Stage	Solution	Solution
AuthorGoals	AUTH0087;AUTH0043;AUTH0044	AUTH0088;AUTH0043;AUTH0044

Table E.6: Modified Theme Entry (Group A)

ThemeID	THME0015
InitActivity	WORD0324 WORD0117 WORD0149
Lesson	WORD0083 WORD0243
MoralLesson	BE BRAVE (school)
RelatedObjects	OB013;OB018;OB028;OB030;OB006;OB012;OB003
Problem	SPAT0026
RisingAction	SPAT0027
Solution	SPAT0028;SPAT0052;SPAT0053
Climax	SPAT0017
InitSettings	WORD0178 WORD0145
InitTime	WORD0147;WORD0150;WORD0018;WORD0153

E.2 Group B Relations

Table E.7: New Lexicon Entries (Group B)

Word	**WORD ID**	**ONTO ID**
oak	WORD0449	ONTO0255
trunk	WORD0450	ONTO0256
root	WORD0451	ONTO0257
back	WORD0452	ONTO0258
head	WORD0453	ONTO0259
leg*	WORD0042	ONTO0260
toe	WORD0454	ONTO0261
person	WORD0455	ONTO0262
book*	WORD0185	ONTO0263
paper	WORD0456	ONTO0264
ink	WORD0457	ONTO0265
glue	WORD0458	ONTO0266
dinner	WORD0459	ONTO0267
eat dinner	WORD0242 WORD0459	ONTO0268
fireman	WORD0460	ONTO0269
librarian	WORD0461	ONTO0270
rescue	WORD0462	ONTO0271
organize	WORD0463	ONTO0272
home*	WORD0352	ONTO0254
dirty*	WORD0077	ONTO0273

Note: *ONLY an entry in the concept table was added. Word is already existing in the lexicon.

Table E.8: New Ontology Entries (Group B)

OntoID	**SemanticRelation**	**Element2**	**Category**
ONTO0002	locationOf	ONTO0033	spatial
ONTO0054	partOf	ONTO0038	things
ONTO0054	partOf	ONTO0254	things
ONTO0262	owns	ONTO0003	generic
ONTO0262	owns	ONTO0070	generic
ONTO0270	roleResponsibleFor	ONTO0272	action
ONTO0269	roleResponsibleFor	ONTO0271	action
ONTO0161	happens	ONTO0043	event
ONTO0268	happens	ONTO0043	event
ONTO0139	happens	ONTO0039	event
ONTO0038	oftenNear	ONTO0055	spatial
ONTO0038	oftenNear	ONTO0058	spatial
ONTO0038	oftenNear	ONTO0057	spatial
ONTO0001	eventForGoalState	ONTO0115	event
ONTO0001	eventForGoalState	ONTO0273	event
ONTO0001	eventForGoalState	ONTO0155	event
ONTO0263	madeOf	ONTO0266	things
ONTO0263	madeOf	ONTO0265	things
ONTO0263	madeOf	ONTO0264	things
ONTO0262	partOf	ONTO0261	things
ONTO0262	partOf	ONTO0260	things
ONTO0262	partOf	ONTO0259	things
ONTO0262	partOf	ONTO0258	things
ONTO0255	partOf	ONTO0257	things
ONTO0255	partOf	ONTO0256	things
ONTO0262	hasRole	ONTO0269	function
ONTO0262	hasRole	ONTO0270	function

Table E.9: New Author Goal Entries (Group B) (Part 1)

GoalID	AUTH0078
Name	Show Set B
Goal	CGOL0038(Agens:WORD0097,Target:%ontoSpatial(WORD0097)%);
Consequence	CGOL0038(Agens:WORD0022,Target:%ontoSpatial(WORD0022)%);
GoalID	AUTH0079
Name	Show partOf
Goal	CGOL0044(Agens:WORD0449,Target:%ontoThings(WORD0449)%);
Consequence	CGOL0044(Agens:WORD0455,Target:%ontoThings(WORD0455)%);
GoalID	AUTH0080
Name	Show madeOf
Goal	CGOL0044(Agens:WORD0185,Target:%ontoThings(WORD0185)%);
Consequence	CGOL0044(Agens:WORD0185,Target:%ontoThings(WORD0185)%);
GoalID	AUTH0081
Name	Show eventForGoalState
Goal	CGOL0038(Agens:WORD0022,Target:%ontoEvent(WORD0022)%);
Consequence	CGOL0038(Agens:WORD0022,Target:%ontoEvent(WORD0022)%);
GoalID	AUTH0082
Name	Show oftenNear
Goal	CGOL0038(Agens:WORD0149,Target:%ontoSpatial(WORD0149)%);
Consequence	CGOL0038(Agens:WORD0149,Target:%ontoSpatial(WORD0149)%);

Table E.10: New Author Goal Entries (Group B) (Part 2)

GoalID	AUTH0083
Name	Show happens
Goal	CGOL0038(Agens:WORD0242 WORD0459, Target:%ontoEvent(WORD0242 WORD0459)%);
Consequence	CGOL0038(Agens:WORD0324 WORD0117 WORD0149, Target:%ontoEvent(WORD0324 WORD0117 WORD0149)%);
GoalID	AUTH0084
Name	Show roleResponsibleFor
Goal	CGOL0006(Agens:WORD0460,Target:%ontoAction(WORD0460)%);
Consequence	CGOL0006(Agens:WORD0461,Target:%ontoAction(WORD0461)%);
GoalID	AUTH0085
Name	Show owns
Goal	CGOL0044(Agens:%job%,Target:%ontoGeneric(WORD0455)%);
Consequence	CGOL0044(Agens:%job%,Target:%ontoGeneric(WORD0455)%);
GoalID	AUTH0086
Name	Shown hasRole
Goal	CGOL0027(Agens:%job%,Target:%ontoFunction(WORD0455)%);
Consequence	CGOL0027(Agens:%job%,Target:%ontoFunction(WORD0455)%);

Table E.11: Modified Story Plot Tracker Entry (Group B)

PlotID	SPAT0004
Name	Learn the benefit
Stage	Climax
AuthorGoals	AUTH0007;AUTH0078;AUTH0079;AUTH0080;AUTH0081; AUTH0082;AUTH0083;AUTH0084;AUTH0085;AUTH0086;AUTH0008

F Generated Stories

The following stories were generated by Picture Books to test the different relations.

Roy the chicken learns to take bath.

The afternoon was fair.
Roy the chicken was in the garden.
He played with a doll. - **usedFor**
Daddy Sam told Roy to take bath.
Roy did not want to take bath.
He continued to play.
Roy did not take the bath.
He became dirty.
Roy felt itchy.
He felt hurt.
Roy cried.
Daddy Sam saw that he was crying.
Daddy Sam told Roy to take bath.
Roy wanted to take bath.
He took the bath with a yellow rubber ducky.
Roy removed dirt.
Taking bath results to smelling nice.
It was fun.
Roy was happy.
Toys were toy store. - **locationOf**
Playing is the garden. - **locationOf**
An oak was a root. - **partOf**
A person was legs. - **partOf**
A book was ink. - **madeOf**
It was ink. - **madeOf**
Playing is healthy. - **evenForGoalState**
It was games. - **eventForGoalState**
The school was the clinic. - **oftenNear**
It was the market. - **oftenNear**
Eating dinner is the evening. - **happens**
Going to the school is the morning. - **happens**
A fireman was rescuing. - **roleResponsibleFor**
A librarian was organizing. - **roleResponsibleFor**
The person was a tricycle. - **owns**

It was the tricycle. - **owns**
From then on, Roy always took the bath.

Roy the chicken learns to take bath.

The day was bright.
Roy the chicken was in the garden.
He played with a tricycle. - **usedFor**
Daddy Sam told Roy to take bath.
Roy did not want to take bath.
He continued to play.
Roy did not take the bath.
He became dirty.
Roy felt itchy.
He felt hurt.
Roy cried.
Daddy Sam saw that he was crying.
Daddy Sam told Roy to take bath.
Roy wanted to take bath.
He took the bath with a yellow rubber ducky.
Roy soothed itchy skin.
Taking bath results to smelling nice.
It was fun.
Roy was happy.
Toys were the bedroom. - **locationOf**
Playing is the playground. - **locationOf**
An oak was trunk. - **partOf**
A person was legs. - **partOf**
A book was paper. - **madeOf**
It was glue. - **madeOf**
Playing is a knick knack. - **eventForGoalEvent**
It was dirty. - **eventForGoalEvent**
The school was the mall. - **oftenNear**
It was the clinic. - **oftenNear**
Eating dinner is the evening. - **happens**
Going to the school is saying goodbye. - **happens**
A fireman was rescuing. - **roleResponsibleFor**
A librarian was organizing. - **roleResponsibleFor**
The person was the tricycle. - **owns**
It was a ball. - **owns**
From then on, Roy always took the bath.

Cathy the cat learns to take bath.

The afternoon was fair.
Cathy the cat was in the garden.
She played with toy car. - **usedFor**
Mommy Sara told Cathy to take bath.
Cathy did not want to take bath.
She continued to play.
Cathy did not take the bath.
She became dirty.
Cathy felt itchy.
She felt hurt.
Cathy cried.
Mommy Sara saw that she was crying.
Mommy Sara told Cathy to take bath.
Cathy wanted to take bath.
She took the bath with a fragrant soap.
Cathy soothed itchy skin.
Taking bath results to smelling nice.
It was fun.
Cathy was happy.
Toys were the bedroom.Playing is toy store.An oak had a root.A person had a toe.
After that day, Cathy always took the bath.

Porky the pig learns to be honest.

The day was fair.
Porky the pig was in the bedroom.
He played near breakable glass of water.
Porky broke glass of water. - **capableOf, isA**
He was worried. - **effectOf**
Daddy Sam saw that glass of water was smashed. - **propertyOf**
Porky told Daddy Sam that Geena the giraffe broke glass of water.
She was sad.
Geena cried.
Porky felt guilty.
Porky told Daddy Sam that he broke glass of water.
Daddy Sam told Porky that he should have been honest.
He apologized to Daddy Sam.
Porky apologized to Geena.
Daddy Sam told Porky to be honest.
Daddy Sam told Porky that being honest is good.

After that day, Porky always was honest.

Edward the elephant learns to be careful.

The afternoon was sunny.
Edward the elephant was in the dining room.
He played near breakable glass of water.
Mommy Edna told Edward to be careful.
He continued to play glass of water near.
Edward fell it. - **capableOf, isA**
Edward was worried. - **effectOf**
He hid away from Mommy Edna.
Mommy Edna saw that glass of water was cracked. - **propertyOf**
Mommy Edna called Edward.
Mommy Edna told Edward that he should have obeyed.
Edward felt sorry.
Mommy Edna cleaned up glass of water.
Edward helped Mommy Edna to clean up.
She reminded him to be careful.
Being careful is important.
Afterwards, Edward always was careful.

Edward the elephant learns to be neat.

The morning was warm.
Edward the elephant was in the garden.
Mommy Edna told Edward that he should not be messy.
Edward wanted to rejoice. - **eventForGoalEvent**
He rejoiced toys. - **eventForGoalEvent**
Toys scattered.
Edward created a mess.
He wanted to find a red ball.
Edward could not find the red ball.
He felt sad.
Edward cried.
Mommy Edna saw that he was crying.
Edward told Mommy Edna that he could not find the ball.
Mommy Edna told Edward that he should have listened.
Mommy Edna helped Edward to clean up.
He found lost toy.
Being neat is good.

Edward was happy.
After that day, he always was neat.

Edward the elephant learns to be neat.

The evening was cool.
Edward the elephant was in the living room.
Mommy Edna told Edward that he should not be messy.
<u>Edward wanted to revel. - **eventForGoalEvent**</u>
<u>He revelled toys. - **eventForGoalEvent**</u>
Toys scattered.
Edward created a mess.
He wanted to find a red ball.
Edward could not find the red ball.
He felt sad.
Edward cried.
Mommy Edna saw that he was crying.
Edward told Mommy Edna that he could not find the ball.
Mommy Edna told Edward that he should have obeyed.
Mommy Edna helped Edward to clean up.
He found lost toy.
Being neat is good.
Edward was happy.
After that day, he always was neat.

Ellen the elephant learns to be brave.

The afternoon was fair.
Ellen the elephant was at the school.
She went with Mommy Edna to the school.
Mommy Edna said to Ellen a goodbye.
Ellen was scared.
She cried.
Mommy Edna saw that Ellen was crying.
Mommy Edna told Ellen to be brave.
She introduced to her a class.
Ellen felt shy.
She wanted to be brave.
Ellen was brave.
She wanted to play with others.
Ellen made friends.
<u>Being brave results to a friendship. - **effectOfIsState**</u>

Ellen had friends.
She had new playmates.
Ellen played games.
She was happy.
Mommy Edna asked Ellen how was everything.
Ellen told Mommy Edna that she felt better.
Being brave is good.
Ellen was happy.
After that day, she always was brave.

Edward the elephant learns to be brave.

The afternoon was fair.
Edward the elephant was at the school.
He went with Mommy Edna to the school.
Mommy Edna said to Edward a goodbye.
He was scared.
Edward cried.
Mommy Edna saw that he was crying.
Mommy Edna told Edward to be brave.
She introduced to him a class.
Edward felt shy.
He wanted to be brave.
Edward was brave.
He wanted to play with others.
Edward made friends.
Being brave results to closeness. - **effectOfIsState**
Edward had friends.
He had new playmates.
Edward played games.
He was happy.
Mommy Edna asked Edward how was everything.
Edward told Mommy Edna that he felt better.
Being brave is good.
Edward was happy.
Afterwards, he always was brave.

Ellen the elephant learns to be brave.

The evening was warm.
Ellen the elephant was at the school.
She went with Mommy Edna to the school.

Mommy Edna said to Ellen a goodbye.
Ellen was scared.
She cried.
Mommy Edna saw that Ellen was crying.
Mommy Edna told Ellen to be brave.
She introduced to her a class.
Ellen smiled.
She wanted to be brave.
Ellen was brave.
She wanted to play with others.
Ellen made friends.
Being brave results to a relationship. - **effectOfIsState**
Ellen had friends.
She had new playmates.
Ellen played games.
She was happy.
Mommy Edna asked Ellen how was everything.
Ellen told Mommy Edna that she felt better.
Being brave is good.
Ellen was happy.
From that day onwards, she always was brave.

G Gold Standard Comparative Results

The following table show the detailed results after comparing the extracted relations to the gold standard

Table G.1: Gold Standard Evaluation Results - Everybody Cries (Raw)

Relations	**Gold Standard**	**Extraction**	**Delta**	**Correct**	**P**	**R**	**F**
Overall	456	392	-64	137	0.35	0.30	0.32
IsA	10	13	3	0	0.00	0.00	0.00
PropertyOf	78	134	56	45	0.34	0.58	0.42
PartOf	9	9	0	4	0.44	0.44	0.44
MadeOf	0	0	0	0	0.00	0.00	0.00
EventForGoalEvent	25	23	-2	4	0.17	0.16	0.17
EventForGoalState	4	3	-1	1	0.33	0.25	0.29
EffectOf	78	64	-14	7	0.11	0.09	0.10
EffectOfIsState	15	0	-15	0	0.00	0.00	0.00
CapableOf	172	76	-96	51	0.67	0.30	0.41
OftenNear	1	0	-1	0	0.00	0.00	0.00
LocationOf	11	7	-4	2	0.29	0.18	0.22
UsedFor	2	5	3	0	0.00	0.00	0.00
Happens	4	10	6	0	0.00	0.00	0.00
HasRole	1	1	0	0	0.00	0.00	0.00
RoleResponsibleFor	12	3	-9	2	0.67	0.17	0.27
Owns	34	44	10	21	0.48	0.62	0.54

Table G.2: Gold Standard Evaluation Results - Everybody Cries (Modified)

Relations	**Gold Standard**	**Extraction**	**Delta**	**Correct**	**P**	**R**	**F**
Overall	456	410	-46	156	0.38	0.34	0.36
IsA	10	11	1	0	0.00	0.00	0.00
PropertyOf	78	149	71	47	0.32	0.60	0.41
PartOf	9	10	1	4	0.40	0.44	0.42
MadeOf	0	0	0	0	0.00	0.00	0.00
EventForGoalEvent	25	24	-1	5	0.21	0.20	0.20
EventForGoalState	4	3	-1	1	0.33	0.25	0.29
EffectOf	78	61	-17	10	0.16	0.13	0.14
EffectOfIsState	15	0	-15	0	0.00	0.00	0.00
CapableOf	172	84	-88	64	0.76	0.37	0.50
OftenNear	1	0	-1	0	0.00	0.00	0.00
LocationOf	11	8	-3	2	0.25	0.18	0.21
UsedFor	2	3	1	0	0.00	0.00	0.00
Happens	4	9	5	0	0.00	0.00	0.00
HasRole	1	1	0	0	0.00	0.00	0.00
RoleResponsibleFor	12	3	-9	2	0.67	0.17	0.27
Owns	34	44	10	21	0.48	0.62	0.54

Table G.3: Gold Standard Evaluation Results - Start School (Raw)

Relations	**Gold Standard**	**Extraction**	**Delta**	**Correct**	**P**	**R**	**F**
Overall	139	150	11	47	0.31	0.34	0.33
IsA	5	2	-3	0	0.00	0.00	0.00
PropertyOf	41	86	45	24	0.28	0.59	0.38
PartOf	0	0	0	0	0.00	0.00	0.00
MadeOf	0	0	0	0	0.00	0.00	0.00
EventForGoalEvent	8	9	1	1	0.11	0.13	0.12
EventForGoalState	0	0	0	0	0.00	0.00	0.00
EffectOf	9	13	4	1	0.08	0.11	0.09
EffectOfIsState	5	0	-5	0	0.00	0.00	0.00
CapableOf	42	24	-18	10	0.42	0.24	0.30
OftenNear	1	0	-1	0	0.00	0.00	0.00
LocationOf	2	1	-1	1	1.00	0.50	0.67
UsedFor	6	1	-5	0	0.00	0.00	0.00
Happens	1	0	-1	0	0.00	0.00	0.00
HasRole	1	1	0	0	0.00	0.00	0.00
RoleResponsibleFor	5	0	-5	0	0.00	0.00	0.00
Owns	13	13	0	10	0.77	0.77	0.77

Table G.4: Gold Standard Evaluation Results - Start School (Modified)

Relations	**Gold Standard**	**Extraction**	**Delta**	**Correct**	**P**	**R**	**F**
Overall	139	173	34	54	0.31	0.39	0.35
IsA	5	2	-3	0	0.00	0.00	0.00
PropertyOf	41	92	51	24	0.26	0.59	0.36
PartOf	0	0	0	0	0.00	0.00	0.00
MadeOf	0	0	0	0	0.00	0.00	0.00
EventForGoalEvent	8	11	3	2	0.18	0.25	0.21
EventForGoalState	0	0	0	0	0.00	0.00	0.00
EffectOf	9	15	6	1	0.07	0.11	0.08
EffectOfIsState	5	0	-5	0	0.00	0.00	0.00
CapableOf	42	36	-6	16	0.44	0.38	0.41
OftenNear	1	0	-1	0	0.00	0.00	0.00
LocationOf	2	1	-1	1	1.00	0.50	0.67
UsedFor	6	2	-4	0	0.00	0.00	0.00
Happens	1	0	-1	0	0.00	0.00	0.00
HasRole	1	1	0	0	0.00	0.00	0.00
RoleResponsibleFor	5	0	-5	0	0.00	0.00	0.00
Owns	13	13	0	10	0.77	0.77	0.77

Table G.5: Gold Standard Evaluation Results - Hopsalot's Garden (Raw)

Relations	**Gold Standard**	**Extraction**	**Delta**	**Correct**	**P**	**R**	**F**
Overall	68	73	5	28	0.38	0.41	0.40
IsA	2	2	0	1	0.50	0.50	0.50
PropertyOf	19	31	12	11	0.35	0.58	0.44
PartOf	0	0	0	0	0.00	0.00	0.00
MadeOf	0	0	0	0	0.00	0.00	0.00
EventForGoalEvent	3	1	-2	0	0.00	0.00	0.00
EventForGoalState	0	0	0	0	0.00	0.00	0.00
EffectOf	8	9	1	1	0.11	0.13	0.12
EffectOfIsState	0	0	0	0	0.00	0.00	0.00
CapableOf	22	16	-6	7	0.44	0.32	0.37
OftenNear	0	0	0	0	0.00	0.00	0.00
LocationOf	0	0	0	0	0.00	0.00	0.00
UsedFor	1	1	0	0	0.00	0.00	0.00
Happens	2	1	-1	0	0.00	0.00	0.00
HasRole	0	0	0	0	0.00	0.00	0.00
RoleResponsibleFor	0	0	0	0	0.00	0.00	0.00
Owns	11	12	1	8	0.67	0.73	0.70

Table G.6: Gold Standard Evaluation Results - Hopsalot's Garden (Modified)

Relations	**Gold Standard**	**Extraction**	**Delta**	**Correct**	**P**	**R**	**F**
Overall	68	79	11	31	0.39	0.46	0.42
IsA	2	2	0	1	0.50	0.50	0.50
PropertyOf	19	33	14	11	0.33	0.58	0.42
PartOf	0	0	0	0	0.00	0.00	0.00
MadeOf	0	0	0	0	0.00	0.00	0.00
EventForGoalEvent	3	4	1	1	0.25	0.33	0.29
EventForGoalState	0	0	0	0	0.00	0.00	0.00
EffectOf	8	7	-1	1	0.14	0.13	0.13
EffectOfIsState	0	0	0	0	0.00	0.00	0.00
CapableOf	22	21	-1	9	0.43	0.41	0.42
OftenNear	0	0	0	0	0.00	0.00	0.00
LocationOf	0	0	0	0	0.00	0.00	0.00
UsedFor	1	1	0	0	0.00	0.00	0.00
Happens	2	0	-2	0	0.00	0.00	0.00
HasRole	0	0	0	0	0.00	0.00	0.00
RoleResponsibleFor	0	0	0	0	0.00	0.00	0.00
Owns	11	11	0	8	0.73	0.73	0.73

References

Agichtein, E., & Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries.*

Alani, H., Kim, S., Millard, D., Weal, M., Lewis, P., Hall, W., et al. (2003). Automatic Extraction of Knowledge from Web Documents. In *Proceedings of ISWC 2003 Workshop on Human Language Technology for the Semantic Web and Web Services.*

Ang, K., Antonio, J., Sanchez, D., & Yu, S. (2010). *Picture Books 2.*

Badulescu, A., Girju, R., & Moldovan, D. (2006). Automatic Discovery of Part-Whole Relations. *Comput. Linguist.*, *32*(1), 83–135.

Banko, M., & Etzioni, O. (2008). The Tradeoffs Between Open and Traditional Relation Extraction. In *Proceedings of ACL Human Language Technology (HLT 2008)* (pp. 23–36).

Berland, M., & Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 57–64). Morristown, NJ, USA: Association for Computational Linguistics.

Cheng, T., Cua, J., Tan, M., & Yao, K. (2008). *Information Extraction for Legal Documents.*

Fillmore, C., Baker, C., & Lowe, J. (1998). The Berkeley Framenet Project. In *Proceedings of the 17th international conference on Computational linguistics* (pp. 86–90). Morristown, NJ, USA: Association for Computational Linguistics.

Gruber, T. (2008). Ontology. In *Encyclopedia of Database Systems.*

Hearst, M. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics* (pp. 539–545).

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Seaghdha, D., Pado, S., et al. (2009). SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009), in conjunction with NAACL-HLT 2009* (pp. 94–99).

Hong, B., & Ong, E. (2009). Automatically Extracting Word Relationships as Templates for Pun Generation. In *Proceedings of the NAACL Human Language Technology 2009 Workshop on Computational Approaches to Linguistic Creativity* (pp. 24–31).

Kipper, K., Dang, H. T., & Palmer, M. (2000). Class-based construction of a verb lexicon. In *Proceedings of Seventeenth National Conference on Artificial Intelligence AAAI 2000.*

Knott, A., & Dale, R. (1994). Using Linguistic Phenomena to Motivate a Set of Rhetorical Relations. *Discourse Processes*, 35–62.

Lester, J., Rowe, J., & McQuiggan, S. (2007). Narrative Presence in Intelligent Learning Environments. In *Association for the Advancement of Artificial Intelligence Symposium on Intelligent Narrative Technologies 2007* (pp. 126–133).

Liu, H., & Singh, P. (2004a). Commonsense Reasoning in and over Natural Language. In *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems* (pp. 293–306).

Liu, H., & Singh, P. (2004b). Conceptnet – A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal*, 211–226.

Mann, W., & Thompson, S. (1987). *Rhetorical Structure Theory: A Theory of Text Organization* (Tech. Rep.). University of Southern California, Marina Del Rey, Information Sciences Institute.

Mateas, M., & Stern, A. (2003). Faade: An Experiment in Building a Fully-Realized Interactive Drama. In *Game Developers Conference, Game Design track.*

Moldovan, D., Badulescu, A., Tatu, M., Antohe, D., & Girju, R. (2004). Models for the semantic classification of noun phrases. In *Proceedings of the Computational Lexical Semantics Workshop at HLT-NAACL 2004* (pp. 60–67).

Montfort, N. (2009). Curveship: An Interactive Fiction System for Interactive Narrating. In *Proceedings of the NAACL Human Language Technology 2009 Workshop on Computational Approaches to Linguistic Creativity* (pp. 55–62).

Mueller, E. (1999). Prospects for In-Depth Story Understanding by Computer. *Journal of Cognitive Systems Research*, 307–340.

Mueller, E. (2003). Story Understanding through Multi-Representation Model. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning* (pp. 46–53).

Mueller, E. (2007). Modelling Space and Time in Narratives about Restaurants. *Literary and Linguistic Computing*, 67–84.

Muslea, I. (1999). Extraction Patterns for Information Extraction Tasks: A Survey. In *Proceedings AAAI-99 Workshop on Machine Learning for Information Extraction* (pp. 46–53).

Nakasone, A., & Ishizuka, M. (2006). Storytelling Ontology Model Using RST. In *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2006)* (pp. 163–169).

Nastase, V., & Szpakowicz, S. (2003). Exploring noun-modifier semantic relations. In *Fifth International Workshop on Computational Semantics (IWCS-5)* (pp. 285–301).

Niles, I., & Pease, A. (2001). Towards a Standard Upper Ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems* (pp. 2–9).

Nodelman, P. (2008). *The hidden adult: Defining children's literature*. Johns Hopkins University Press. Available from http://books.google.com.ph/books?id=wKosfJZhUwQC

Riedl, M., & Young, R. M. (2004). A Planning Approach to Story Generation for History Education. In *Proceedings of the 3rd International Conference on Narrative and Interactive Learning Environments.*

Rosario, B., & Hearst, M. (2001). Classifying the semantic relations in noun-compounds via a domain-specific lexical hierarchy. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-01)* (pp. 82–90).

Rosario, B., Hearst, M., & Fillmore, C. (2002). The descent of hierarchy, and selection in relational semantics. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)* (pp. 417–424).

Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., & Zhu, W. L. (2002). Open Mind Common Sense: Knowledge Acquisition from the General Public. In *Odbase02.*

Solis, C., Siy, J. T., Tabirao, E., & Ong, E. (2009). Planning Author and Character Goals for Story Generation. In *Proceedings of the NAACL Human Language Technology 2009 Workshop on Computational Approaches to Linguistic Creativity.*

Speer, R., & Havasi, C. (2012, may). Representing general relational knowledge in conceptnet 5. In N. C. C. Chair) et al. (Eds.), *Proceedings of the eight international conference on language resources and evaluation (lrec'12).* Istanbul, Turkey: European Language Resources Association (ELRA).

Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., & Soderland, S. (2007). Textrunner: Open information extraction on the web. In *NAACL '07: Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations on XX* (pp. 25–26). Morristown, NJ, USA: Association for Computational Linguistics.

Young, R. M. (2008). Computational Creativity in Narrative Generation: Utility and Novelty Based on Models on Story Comprehension. In *Proceedings of the Association for the Advancement of Artificial Intelligence 2008 Sprint Symposium.*