Creating Logical Representations of Meaning from Plain-Text Stories

A LITERATURE REVIEW SUBMITTED TO THE GRADUATE DIVISION OF THE

DEPARTMENT OF INFORMATION AND COMPUTER SCIENCE AT THE

UNIVERSITY OF HAWAIʻI AT MĀNOA IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR PHD PORTFOLIO

December 11, 2022

By

Bernadette J. Tix

Keywords: Natural Language Processing, Computational Linguistics, Story Models

CONTENTS

ABBREVIATIONS

AI	Artificial Intelligence
ATN	Augmented Transition Network
BNF	Backus-Nuar Form
CFG	Context-Free Grammar
DCG	Definite Clause Grammar
HMM	Hidden Markov Model
LSTM	Long Short-Term Memory (A type of Neural Network)
MemNN	Memory Network
MemN2N	Recurrent Memory Network
NER	Named Entity Recognition
NLP	Natural Language Processing
QA	Question-Answering software
QKS	QUEST Knowledge Structure
RNN	Recurrent Neural Network
RRG	Role and Reference Grammar

FIGURES

1. INTRODUCTION

Stories are one of the primary ways in which human beings think and communicate about our lives and about the world. It has been argued that storytelling contributed to human evolutionary success [6] and that *"narrative is the primary way that we structure experiences into an understandable reality and that our ability to perceive the actual world is based on a general ability to imagine possible worlds"* [7,18]. Because narrative structure is so central to humans' communication and thought process, it would be valuable to have an AI that could parse stories and draw facts and insights from narrative-form text. Several systems exist that can draw useful information from stories, including systems which can answer reading-comprehension questions [29], produce and evaluate summaries [5,34], and extract key phrases that describe the themes or topics of a piece [54].

However, how to automatically parse and model the full content of a story remains an unsolved problem. There are several major difficulties in the task of modelling even the shallowest and most literal meaning of the sentences in a story. Ambiguity in word meaning, sentence structure, and pronoun references makes it impossible for any system to be 100% accurate, and even the best systems currently fall far short. This is a core challenge for any **Natural Language Processing (NLP)** system.

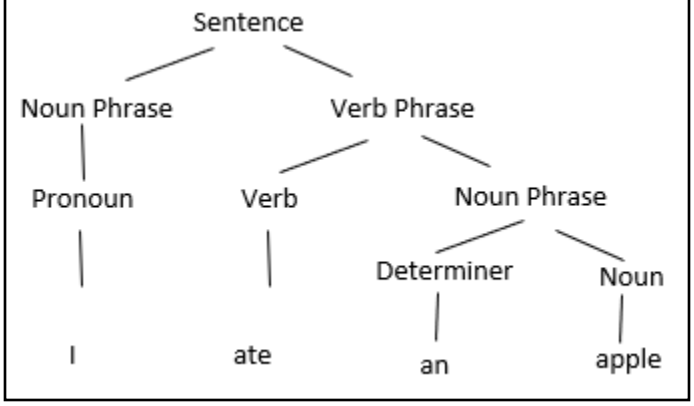
In some cases, sub-symbolic systems can bypass the need for a symbolic representation of meaning altogether. This approach is how tasks such as question-answering and key-phrase extraction are accomplished. However, there are many tasks for which a symbolic representation is still required. Mueller (2003) describes a system capable of determining facts that a reader would infer from a text but which are not explicitly stated. However, this system requires a symbolic representation of the story's explicit text as a starting point. Since this cannot be automatically generated, the story needs to be hand-translated into a series of event calculus clauses by a human being before the system can function [39]. Buscaroli et al. (2022) argue that tasks requiring complex reasoning, rather than just perception or classification, may always require a symbolic, logical model of the problem for at least some stage of the reasoning process [9].

This literature review will cover a wide array of work related to this topic, starting with fundamental works in syntax, named entity recognition, and semantics. Sub-symbolic fact-extraction in the form of question-answering systems and models for representing entire stories will also be covered, as well as a short list of available datasets and open challenges.

2. SYNTAX AND SEMANTICS

Syntax refers to *"the study of the rules for the formation of grammatical sentences in a language."* [55] Software systems that attempt to create meaning from written natural language sources often begin by performing a syntactic parse, which determines the grammatical structure of each sentence. This is commonly done by identifying **Lexical Categories**, which are parts of speech such as *nouns* and *verbs*, and **Syntactic Categories**, which are multi-word phrases such as *noun phrases* and *verb phrases* [46]. The combination of lexical and syntactic categories within a sentence forms a tree representing the **Phrase Structure** of that sentence. A **Phrase Structure Grammar** is a defined set of rules which match specific phrase structures. Ideally, a complete phrase structure grammar will be able to correctly parse as many grammatically correct sentences as possible. According to the **Chomsky Hierarchy**, grammars can be divided into one of four classes, arranged by their generative capacity, where each class has the power to describe all languages described by any less powerful class, as well as some additional languages [13,46]. The four classes are:

Figure 1: Sample Phrase Structure Tree



- **Recursively Enumerable Grammars** have unrestricted rules. Both sides of any rule can have any number of terminal and nonterminal symbols. *Example: A B C -> D E*
- **Context-Sensitive Grammars** require that the right side of the rule contain at least as many symbols as the left side. *Example: A X B -> A Y B*
- **Context-Free Grammars (CFG)** require that the left side of the rule consists of a single non-terminal symbol. CFGs are popular for both natural-language and programming-language grammars [46]. *Example: A -> B C D*
- **Regular Grammars** consist of a single non-terminal symbol on the left side of each rule, and a single terminal symbol optionally followed by a single non-terminal symbol on the right.

The phrase structure tree shown in Figure 1 could have been generated from a CFG, but not from a regular grammar, since the *sentence*, *noun phrase*, and *verb phrase* symbols all match sequences of multiple non-terminal symbols.

While syntax is the study of a sentence's structure, **Semantics** is the study of that sentence's meaning. There are various ways of representing the meaning of a sentence, including **Frames** [15], **Semantic Roles** [21], **Semantic Domains** [22]**,** and **Event Calculus** [40]**.** This section will explore several methods for representing and extracting semantic meaning from text.

2.1 *Role and Reference Grammar*

Robert D. Van Valin Jr., 1993 [50]

Role and Reference Grammar (RRG) is a linguistic theory of clause construction across multiple languages. Though not specifically geared towards the development of AI software systems, it is still relevant as a linguistic theory of grammar. RRG is a *"structural-functionalist theory of grammar,"* meaning that it is concerned with both the structural elements of syntax as well as the functions that grammar serves in facilitating human communication, and thus differs from purely formalist or purely functionalist approaches to explaining syntax and semantics. RRG originated as a divergence from earlier theories which were overly focused on English at the expense of grammar structures found in other languages.

> *"RRG grew out of an attempt to answer two fundamental questions: (1) what would linguistic theory look like if it were based on the analysis of Lakhota, Tagalog, and Dyirbal, rather than on the analysis of English?; and (2) how can the interaction of syntax, semantics, and pragmatics in different grammatical systems best be captured and explained?"* [50]

RRG eschews standard formats for explaining clause structure, since syntax varies from language to language and thus any model based on a specific clause structure will necessarily impose some of the syntax of whatever language the model originated from. Instead, RRG defines clauses in terms of a layered structure, including a nucleus, which includes the predicates of the clause, a core, which includes the nucleus plus the arguments of the predicates, and the periphery, which includes modifiers to the core. These three layers are universal across languages. Some languages also contain unique layers in addition to these three.

The RRG model is primarily concerned with the semantic relationships between different parts of a sentence, and how these relationships can be defined in language-independent ways. It posits that in complex sentences, clauses are related to each other in one of three ways: coordination, subordination, and co-subordination, which is a form of dependent coordination. The following examples are provided for each:

- Coordination: Fred talked to Mary, and she agreed to his suggestion.
- Subordination: Max called Sue, because he was going to be late for the party.
- Co-subordination: Having called Sue, Max left for the party.

Verb phrases are categorized as states, achievements, accomplishments, and activities. The following examples are provided for verb phrase categories:

- State: The lamp is broken.
- Achievement: The lamp broke.
- Accomplishment: Bill broke the lamp.
- Activity: The lamp is shaking.

Noun phrases are broken up into subject and object phrases and are further classified as being an ACTOR or UNDERGOER. The following examples are provided:

- The boy [SUBJ, ACTOR] ate the sandwich [OBJ, UNDERGOER].
- The sandwich [SUBJ, UNDERGOER] was eaten by the boy [ACTOR].
- The girl [SUBJ, ACTOR] ran down the stairs.
- The girl [SUBJ, UNDERGOER] got sick.

From here, the subject and object can be further classified into more specific categories including agent, effector, experiencer, locative, theme, or patient.

This paper was published in the same year as a book by the same author which goes into much greater detail on RRG [51].

2.2 *Automatic Labeling of Semantic Roles*

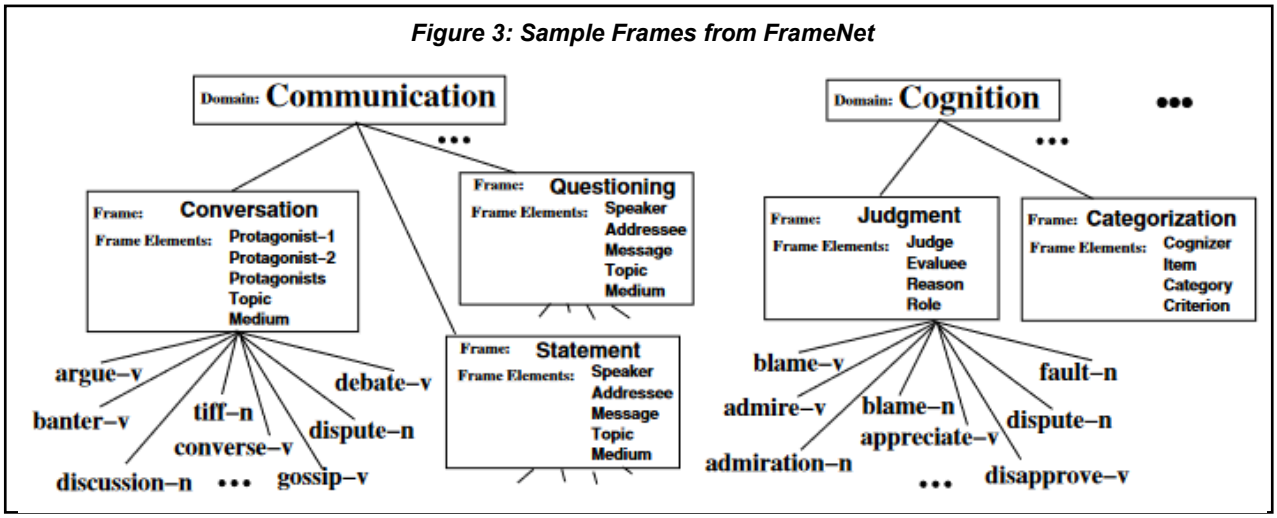
Daniel Gildea and Daniel Jurafsky. 2002. [21]

A **shallow semantic parse** of a sentence identifies the **semantic roles** filled by the constituents of a sentence within a **semantic frame**. Semantic roles are one of the oldest constructs within linguistic theory, dating back thousands of years and with a wide variety of theories of semantic roles having been developed at different times and places throughout history. In modern history, linguists have tended to prefer theories with a small number of highly abstract semantic roles, in some cases as few as two roles, such as *proto-agent* and *proto-patient*. Many more theories use approximately 10 roles, still highly abstract but not quite as extreme. Computer scientists have tended to prefer less abstract, highly specific roles when creating language parsers that handle only very specific classes of text, such as queries about airline tickets including roles such as *FROM_AIRPORT, TO_AIRPORT*, and *DEPART_TIME*. The system presented here seeks a useful middle ground, using 67 frame types categorized into 12 general semantic domains. Figure 3 lists the semantic domains with a partial listing of the frames in each domains, and example predicates. The list of frames and their descriptions was taken from the FrameNet project [30], and frames were further hand-annotated by the team for easier use within their system. The *Judgement* frame is one example, which contains roles like *judge, evaluee,* and *reason,* with a sample sentence provided of *"[Judge She] blames [Evaluee the Government] [Reason for failing to do enough to help] ."* The *statement* frame contains roles like *speaker, addressee,* and *message*, such as "*[Message "I'll knock on your door at quarter to six"] [Speaker Susan] said."* In addition to frame elements, each frame defines predicates along with a statistical liklihood that any given predicate corresponds to the frame in question. The structure of a few FrameNet frames is shown in Figure 4.

Figure 2: Semantic Domains with Sample Frames and Predicates

Domain	Sample Frames	Sample Predicates
Body	Action	flutter, wink
Cognition	Awareness	attention, obvious
	Judgment	blame, judge
	Invention	coin, contrive
Communication	Conversation	bicker, confer
	Manner	lisp, rant
Emotion	Directed	angry, pleased
	Experiencer-Obj	bewitch, rile
General	Imitation	bogus, forge
Health	Response	allergic, susceptible
Motion	Arriving	enter, visit
	Filling	annoint, pack
Perception	Active	glance, savour
	Noise	snort, whine
Society	Leadership	emperor, sultan
Space	Adornment	cloak, line
Time	Duration	chronic, short
	Iteration	daily, sporadic
Transaction	Basic	buy, spend
	Wealthiness	broke, well-off

Figure 3: Sample Frames from FrameNet



A learning algorithm which builds a statistical model to classify any sentence into one of the defined semantic frames is also presented, with roles broken down as shown in the sample sentences above. The system was trained on roughly 50,000 sentences hand-annotated by FrameNet. Features of each sentence are extracted through a syntactic parse that produces a parse tree similar to that shown in Figure 1. The tree hierarchy is used to encode the syntactic relationship of each constituent, and all of this information is used in the evaluation of the sentence. This system was able to achieve an overall accuracy of 82% in identifying the roles of pre-segmented constituents, but only 65% precision and 61% recall at the more difficult task of simultaneously segmenting constituents and identifying semantic roles within unsegmented plain-text sentences.

2.3 *Semantic Domains and Linguistic Theory*

Gliozzo, 2006. [22]

The theory of **Semantic Domains** is an expansion to the older theory of **Semantic Fields** [35]. The theory of semantic fields posited that word senses can be grouped into closely related categories called semantic fields, such as “blue” being within the semantic field of “color.” It has been shown that semantic fields map closely between different languages, even when individual words within those fields do not. For instance, one can find the field of “color” in every language, even though different languages make different delineations between different colors, and there is not always a one-to-one translation between specific color words. The main limitation of semantic field theory is that it does not provide any objective criteria to identify and delineate fields.

Semantic domains theory seeks to improve upon semantic fields by drawing on the insight of Ludwig Wittgenstein that *“Meaning is Use”* [53]. According to Wittgenstein, all language is a form of linguistic game in which the meaning of words depends on the context of their use. Gliozzo provides the example of the word *virus* as used in the domain of Biology, which has a different meaning when used in the domain of Computer Science. By identifying words from the same domain in close proximity we can both predict the domain the text is taking place within and clarify ambiguity in the meaning of the words themselves. For example, if one encounters the word *fork*, this could be a utensil, a fork in the road, a fork in a multi-threaded computer program, or other uses depending on context. However, by noting that the within a short space a text mentions a *fork,* a *spoon,* a *glass*, and a *napkin*, we can have much greater confidence that the meaning of fork in this context is a utensil.

Semantic domains are best delineated as “*…common areas of human discussion, such as ECONOMICS, POLITICS, LAW, SCIENCE, which demonstrate lexical coherence.*” The WordNet Domains project [36] has made significant progress in adding domain labels to WordNet [38] in an effort to make a large database of domain-labelled vocabulary available for use in future research. The authors have demonstrated the effectiveness of domains as a tool for disambiguation in related prior works [27,37]. Domain information has been shown to be critical in the task of word-sense disambiguation, and domain-based approaches achieve state-of-the-art performance in this task [23]. Domain-model based approaches have also had success in the areas of text categorization [25], term categorization [16], ontology learning [24], and multilinguality [26].

2.4 *Syntax and Semantics Discussion*

There are several key difficulties when parsing the syntax and semantics of plain-text data:

- Sentence structure is highly variable, making it impractical to create a CFG which encompasses all possible sentences.
- The meaning and lexical category of individual words can vary based on context, which can create difficulties for parsing both syntax and semantics. For example, *fork* can be a noun or a verb depending on the context of its use, which makes not only its meaning but also its role in the sentence structure ambiguous. This ambiguity needs to be resolved to accurately parse a sentence.
- Techniques which work in one language may not be applicable across all languages.
- Theories of syntax and semantics that work well in the humanities do not always transfer well to applications in computer science. Since humans are used to dealing with a great deal of ambiguity in language, theories intended for use by linguists are rarely precise enough to be implemented as a computer program, unless they have been specifically formulated with this level of precision in mind.

Even the best automatic parsers currently available fall far short of 100% accuracy. *Automatic Labeling of Semantic Roles*, discussed in this section, reported an accuracy of 82% when working with text that had been specially prepared, but only 65% when working with unprepared text. Since language parsing tends to be a multi-step process, even a relatively small percentage of inaccuracy in an early parsing step can result in cascading inaccuracies throughout the system.

There is a wealth of work on the topics of syntax and semantics from both the field of linguistics and the field of computer science. However, fully automatic parsing of plain text in English or any other language is a problem which is still not fully solved. Although systems exist which can parse sentences with some degree of accuracy, the accuracy of these systems is low enough to keep them from being fully effective tools for natural language processing.

3. NAMED ENTITY RECOGNITION

Named Entity Recognition (NER) is the task of identifying and labelling people, places, organizations, locations, and other named entities within a text. While the syntax of a grammar identifies the structure of the sentence including noun phrases, verb phrases, and other parts of speech, it is only through NER that a system can identify what the various noun phrases represent. In the context of a story, a named entity could be a character, a location, or a significant object.

3.1 *Design Challenges and Misconceptions in Named Entity Recognition*

Lev Ratinov and Dan Roth. 2009 [43]

Figure 4: Example of Ambiguous Entity Names [4]

SOCCER - [PER BLINKER] BAN LIFTED .
[LOC LONDON] 1996-12-06 [MISC Dutch] forward [PER Reggie Blinker] had his indefinite suspension lifted by [ORG FIFA] on Friday and was set to make his [ORG Sheffield Wednesday] comeback against [ORG Liverpool] on Saturday . [PER Blinker] missed his club's last two games after [ORG FIFA] slapped a worldwide ban on him for appearing to sign contracts for both [ORG Wednesday] and [ORG Udinese] while he was playing for [ORG Feyenoord].

There are several fundamental difficulties with the task of **Named Entity Recognition (NER),** and various techniques have been developed for overcoming these difficulties. Figure 4 shows an example of a block of text from a piece of sports news, which has been annotated by an NER process. In this case, "Blinker" is the name of an athlete, and "Wednesday" is the name of an organization. This is obviously not the normal use of these words, and their intended meaning is only clear in context. Because of difficulties of this sort, NER is highly dependent on prior knowledge and NER systems perform significantly better when paired with a knowledge base considering non-local features.

There are four key design decisions in the implementation of an NER system:

1. How to represent text chunks
2. What interference algorithm to use
3. How to model non-local dependencies
4. How to use external knowledge resources.

Different representations and algorithms can be compared by testing the algorithms against the same datasets and measuring their performance. In this case, the CoNLL-2003 shared task data [47], MUC7 data [56], and a collection of 20 manually selected websites with diverse content are used to measure the performance of the top systems. The study first examines the two most common schemes for text-chunk representation, ***B**eginning, **I**nside, **O**utside (**BIO**)*, and ***B**eginning, the **I**nside and the **L**ast tokens of multi-token chunks as well as **U**nit-length chunks (**BILOU**).* BILOU consistently outperforms the more common BIO scheme by 1-2% across tests in each of the datasets. Several mechanisms for including non-local

knowledge are compared, such as unlabeled bodies of text and dictionary-like structures called gazetteers gathered from various sources including Wikipedia.

Named entities in the beginning of documents tend to be more easily identifiable and match gazetteers more often. Because of this, NER accuracy can be improved by each individual NER classifications taking into consideration prior classifications that were made earlier in the text. One problem with this technique is that the use of a specific word may not always signify the same entity. For example, a news story that first mentions *Australia* (the country) and later mentions *The Bank of Australia* (an organization).

Multiple non-local features were tested for positive impact on NER, including *context aggregation,* which considers the context from other appearances of a term within the text, *extended prediction history,* which takes advantage of NER classifications earlier in the document being more accurate than those later in the document, and *two-stage prediction aggregation*, which uses a second pass of NER classification that uses classifications made with high confidence anywhere in the document to make better predictions about classifications made with low confidence elsewhere in the document. The use of external knowledge bases was also tested, including unstructured text, word class models, and gazetteers. With all techniques applied, the system was able to achieve an accuracy of 74.53% when using the least structured test set, the 20 selected webpages, but up to 90.8% accuracy on the CoNLL03 test data.

3.2 *An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition*

Vijay Krishnan and Christopher D. Manning. 2006. [31]

This paper demonstrates the use of a two-stage NER process to produce state-of-the-art (at the time of publication) accuracy and performance. In the first stage, NER labelling is performed using only local features. The second stage makes use of both the unlabeled source text and the results of the first-stage NER results. The result is a system which is both faster and more accurate than previously implemented NER systems.

Sequence models such **as Hidden Markov Models (HMM)** cannot capture non-local features within source text. An example of a non-local feature is a previous reference to an entity that is significantly removed from the next reference to that entity. The goal of label consistency ought to consider subsets of previous labels as likely referring to the same entity. For example, a text that references *Albert Einstein* in one sentence, and just *Einstein* in a later sentence, is likely referring to the same person.

The system uses a **Conditional Random Field (CRF)** as the basis for both NER stages, which is considered to be the state of the art for NER inference [32,48]. The performance of the two-stage system is heavily dependent on the performance of the baseline NER system that is used in the first stage and expanded upon in the second stage. The authors describe their baseline NER system as follows:

> *"We use features that have been shown to be effective in NER, namely the current, previous and next words, character n-grams of the current word, Part of Speech tag of the current word and surrounding words, the shallow parse chunk of the current word, shape of the current word, the surrounding word shape sequence, the presence of a word in a left window of size 5 around the current word and the presence of a word in a left window of size 5 around the current word. This gives us a competitive baseline CRF using local information alone[.]"*

The second stage takes advantage of six types of additional features which are produced as outputs of the first stage. These features are Document-level Token-majority features, Document-level Entity-majority features, Document-level Superentity-majority features, Corpus-level Token-majority features, Corpus-level Entity-majority features and Corpus-level Superentity-majority features.

The system was tested against the CoNLL-2003 data set [47] with a training set consisting of 945 documents and approximately 203,000 tokens. The system was found to produce an accuracy of 87.24% correct labelling of entities, which is more accurate than prior works. The system was also able to achieve this accuracy with a shorter runtime when compared with prior works [8,19].

3.3 *Neural Architectures for Named Entity Recognition*

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [33]

Many of the best NER systems rely on large corpora of labelled external data to train the NER algorithm. However, large corpora of labeled data (such as the gazetteers used in Krishnan and Manning 2006) are not readily available in many languages other than English and are labor-intensive to produce. Therefore, an NER algorithm that is designed to operate without any reliance on a large body of labelled data has the potential to be more effective across multiple languages. The task for this system is to identify which words or sets of words are names, bearing in mind that a name is often a multi-word phrase. Tests for this system were carried out in four languages: English, Dutch, Spanish, and German, using collections of unlabeled texts from each language as test data.

Since names often consist of multiple words, it is important for the system to reason jointly over multiple words when making its classification. Additionally, evidence suggesting that a particular word or sequence is a name must include both orthographic evidence (how the word is spelled) and distributional evidence (where the words appear in the text). Word representations within this model thus include both character-based word representations and distributional information.

Recurrent Neural Networks (RNN) are a type of neural network designed to analyze sequential data. However, although RNNs can in theory remember and account for features which occur far apart from each other in a sequence, in practice RNNs prioritize new information over old information and often lose information about distant features. **Long Short-Term Memory (LSTM)** is a response to this weakness of RNNs. An LSTM is an RNN that incorporates a memory cell that enables longer retention of distant features.

The NER system presented in this paper uses a layered architecture including multiple layers of LSTM networks. Word representations are vectorized and used as inputs to two layers of LSTM networks. One LSTM analyzes the data in the forward direction, the other analyzes the data in reverse. Since the LSTM prioritizes new information over old information, the forward-analyzing LSTM will more accurately capture information from the end of each word, and the reverse-analyzing LSTM will more accurately capture information from the beginning of each word. The results from both LSTMs are then combined using a CRF layer to produce final tagging decisions.

In all four languages, this system was shown to be competitive with prior work that relied on external labeled data. In the English and German tests, the system approaches but cannot quite match the best performance of NER systems using external labelled data. However, it exceeds the performance of all prior systems which did not use external labelled data. In the Spanish and Dutch tests, the system surpasses the best performing prior works both with and without external labelled data.

3.4 *Named Entity Recognition Discussion*

Named Entity Recognition faces similar challenges as syntax and semantics. As with syntax and semantics, techniques that work well in English are not always practical in other languages, though in this case the issue is mostly due to a lack of structured training data resources in languages other than English. Ambiguity is a problem as well. An accurate NER system must be able to identify not only individual words, but also multi-word phrases that refer to a single entity. Additionally, named entities may not always be referred to with the same word or phrase, such as the example of a document that first references *Albert Einstein* and later references just *Einstein*. There is also the problem that the same word may be used to refer to multiple different entities. The same document that refers to *Albert Einstein* and *Einstein* may later refer to *Einstein Bros. Bagels,* a restaurant chain that serves bagels and has no relation to the famous physicist [57].

As with syntax and semantic parsing, NER suffers from a lack of accuracy in even state-of-the-art systems. The system presented in *Krishnan and Manning* (2006) achieved an accuracy of 87.24%. The system presented in *Ratinov and Roth* (2009) achieved accuracies between 74.53% up to 90.8%. However, the 90.8% was only achieved using structured test data. The less structured the input, the lower the accuracy.

An additional challenge with NER is that most systems rely on large corpora of structured test data, which is not available in all languages, and which may hamper the development of NER systems even in English if no body of training data is available that closely matches the needs of any given system, though *Lample et al.* (2016) has provided a solution that overcomes that particular hurdle.

In the context of understanding fictional stories, NER may be even more difficult. Names of people and places may not reference real-world people or places and may even be composed of nonsense words or names entirely unique to that particular story. An ideal NER for story interpretation should be able to identify names from their structure and context alone, which may preclude the use of some of the most effective NER techniques. This is another difficulty that might be resolved by *Laple et al.* (2016).

4. QUESTION-ANSWERING SYSTEMS

In recent years there has been considerable advancement in **Question Answering (QA)** systems, AI systems that can answer questions expressed in natural language using data from a large body of text or a more structured knowledgebase. Each of the works in this section describe different designs for this type of question-answering system. Some of these rely on multiple-choice answers being provided as input, and the system selects the best possible answer to the question from the options provided. Others can provide one-word or short answers to simple queries without being given options to choose between.

One key distinction between a question-answering system and a parser or NER is that a parser or NER will attempt to interpret and label all the text that it has been provided with. A parser will attempt to determine the syntactic structure and semantic meaning of every sentence, while NER will attempt to label all recognized named entities within a text. By contrast, a question-answering system is only concerned with identifying a single specific answer from a large body of text. Any information not relevant to the question the system is attempting to answer can be discarded or ignored, and the final output is a single short answer, not an interpretation of the text as a whole.

4.1 *Large-scale Simple Question Answering with Memory Networks*

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015 [4]

The task of **Simple Question Answering** is to answer questions for which the reasoning required is simple so long as one can retrieve the correct evidence for a given question, which can be difficult in large-scale conditions. There is a long history of QA systems, but most recent works have focused on answering questions of increasing sophistication, which require combining information from multiple facts to find the correct answer. However, the problem of answering simple questions requiring only one fact from a knowledge base is still far from solved. Most previous QA systems have been restricted to a few question templates tested against relatively small sets of benchmark questions. In response to these shortcomings, Bordes *et al* have collected a large-scale dataset called **SimpleQuestions** containing over 100,000 human-written questions, each associated with a single fact from the Freebase knowledgebase [2].

Facts within the knowledgebase are structured as *(subject, relationship, object)*, so any question which can be phrased as *(subject, relationship, ?)* should be trivial to answer if a corresponding fact is found which matches the subject and relationship. However, the goal of the system is to answer questions provided in natural language, not questions formatted as *(subject, relationship, ?)*. Questions are converted from natural language into a n-gram of words representation, and from these n-grams candidate facts from the knowledgebase are identified based on which facts with subjects that match one of the words in the question. By re-arranging the words in the question n-gram, the system checks different possible formats of the question against the answer provided by possible facts within the knowledgebase. Each candidate fact is then scored based on how closely the question n-gram and answer n-gram match. A **Memory Networks (MemNN)** [49] module then selects between the top candidate facts to generate the system's final response. The system is built using MemNNs with the hopes that this will allow the system to be scaled to more complex tasks in the future.

Both the candidate scoring and the final response portions of the system were trained using a combination of question-answer pairs from SimpleQuestions, WebQuestions [1], and questions automatically generated from Freebase. The system was tested against questions from SimpleQuestions, WebQuestions, and ReVerb [17] and was able to achieve state-of-the-art (at the time of publication) performance when compared to prior works tested using the WebQuestions question set. The performance of the system was competitive with prior works tested against the ReVerb question set but did not achieve the best performance overall, achieving an accuracy of 68% compared to the best prior system which had an accuracy of 73%.

4.2 *The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations*

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. [29]

The **Children's Book Test** is a new question set for NLP systems. It has been built using freely available children's literature from Project Gutenberg [58]. Children's story books were chosen due to having clear narrative structure, which makes context both clearer and more important to the interpretation of any given sentence. The question set is formed by taking 21 consecutive sentences from the chapters of selected stories. The first 20 sentences form the context for the question, and the 21st sentence is turned into a question by removing one word from the sentence. The task for the system is to determine what the missing word should be. An example is shown in Figure 5 below.

Figure 5: Sample Context and Missing Word Question

"Well, Miss Maxwell, I think it only fair to tell you that you may have trouble with those boys when they do come. Forewarned is forearmed, you know. Mr. Cropper was opposed to our hiring you. Not, of course, that he had any personal objection to you, but he is set against female teachers, and when a Cropper is set there is nothing on earth can change him. He says female teachers can't keep order. He 's started in with a spite at you on general principles, and the boys know it. They know he'll back them up in secret, no matter what they do, just to prove his opinions. Cropper is sly and slippery, and it is hard to corner him."

"Are the boys big ?" queried Esther anxiously.

"Yes. Thirteen and fourteen and big for their age. You can't whip 'em -- that is the trouble. A man might, but they'd twist you around their fingers. You'll have your hands full, I'm afraid. But maybe they'll behave all right after all."

Mr. Baxter privately had no hope that they would, but Esther hoped for the best. She could not believe that Mr. Cropper would carry his prejudices into a personal application. This conviction was strengthened when he overtook her walking from school the next day and drove her home. He was a big, handsome man with a very suave, polite manner. He asked interestedly about her school and her work, hoped she was getting on well, and said he had two young rascals of his own to send soon. Esther felt relieved. She thought that Mr. Baxter had exaggerated matters a little.

S: 1 Mr. Cropper was opposed to our hiring you .
2 Not , of course , that he had any personal objection to you , but he is set against female teachers , and when a Cropper is set there is nothing on earth can change him .
3 He says female teachers ca n't keep order .
4 He 's started in with a spite at you on general principles , and the boys know it .
5 They know he 'll back them up in secret , no matter what they do , just to prove his opinions .
6 Cropper is sly and slippery , and it is hard to corner him . ''
7 `` Are the boys big ? ''
8 queried Esther anxiously .
9 `` Yes .
10 Thirteen and fourteen and big for their age .
11 You ca n't whip 'em -- that is the trouble .
12 A man might , but they 'd twist you around their fingers .
13 You 'll have your hands full , I 'm afraid .
14 But maybe they 'll behave all right after all . ''
15 Mr. Baxter privately had no hope that they would , but Esther hoped for the best.
16 She could not believe that Mr. Cropper would carry his prejudices into a personal application .
17 This conviction was strengthened when he overtook her walking from school the next day and drove her home .
18 He was a big , handsome man with a very suave , polite manner .
19 He asked interestedly about her school and her work , hoped she was getting on well , and said he had two young rascals of his own to send soon .
20 Esther felt relieved .

q: She thought that Mr. _____ had exaggerated matters a little .

C: Baxter, Cropper, Esther, course, fingers, manner, objection, opinion, right, spite.

a: Baxter

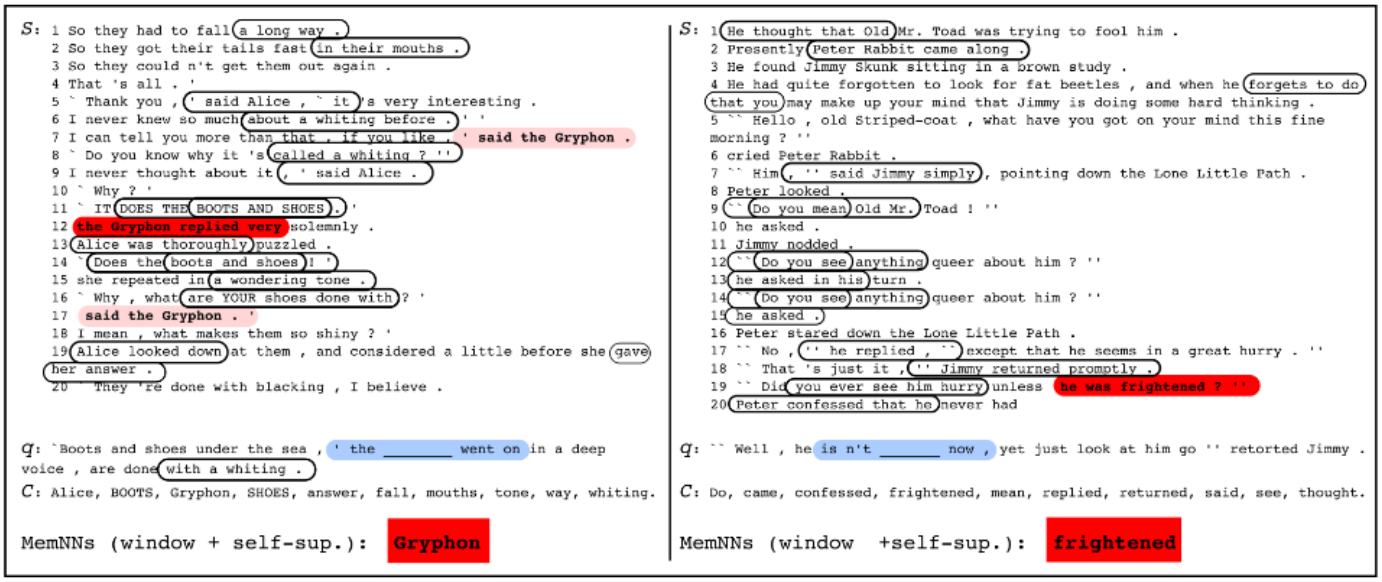
Four classes of question were evaluated, each removing a different kind of word: Named Entities, Common Nouns, Verbs, and Prepositions. For each question nine incorrect answers were selected at from words in the context with the same type as the correct answer.

The authors also developed a new architecture for a system to answer these questions, using a recurrent Memory Network called MemN2N, which allows for direct training of the Memory Network through backpropagation. The use of Memory networks to answer language questions was previously shown in Bordes et. Al. (2015) [4], and this work is an expansion of the capabilities shown in that one. The MemN2N system was compared to a competing system designed using Contextual LSTM, the previous state-of-the-art, and to the performance of 15 human participants who were each given a random sampling of 10% of the total question set. The MemN2N system was found to perform better than the

LSTM system when the missing word was a common noun or named entity but was worse than the LSTM when identifying verbs or prepositions. The human participants performed significantly better than either system when the missing word was a common noun or named entity but were comparable to the LSTM when the missing word was a verb. Humans performed worse than the LSTM when the missing word was a preposition.

The title of the paper, "*The Goldilocks Principle,*" comes from the discovery that the way the context was encoded into the memory of the LSTM had a substantial impact on the performance of the system. While encoding each word separately provided good performance for verbs and prepositions, the performance for nouns and named entities was improved by encoding the context as multi-word phrases. Optimal performance was found when encoding more than one word but less than a complete sentence, with "memory window" size of 5 found to be optimal, centered on the candidate word. Figure 6 below provides an example illustrating the use of memory windows. One major drawback of this system is that the way the experiment is set up, the correct answer is guaranteed to be among the ten candidates, so the system only needs to compare 10 5-word windows. The paper does not include a mechanism for selecting candidates from text when the correct answer is not known during the setup phase.

Figure 6: Sample Memory Windows



4.3 *DramaQA: Character-Centered Video Story Understanding with Hierarchical QA*

Seongho Choi, Kyoung-Woon On, Yu-Jung Heo, Ahjeong Seo, Youwon Jang, Minsu Lee, and Byoung-Tak Zhang. 2021. [12]

This paper differs from the other works described in this section. While it provides a description of a system designed to answer questions about a narrative, in this case the narrative is in the form of video clips from the Korean TV drama *Miss Oh.* The video dataset has been highly edited, reducing the film to only 3 frames per second, with multiple human-added annotations including bounding boxes identifying actors' bodies and faces as well as tags indicating the character's actions and emotional state. These annotations were included in both the video and the script, both of which are used as input data to the **DramaQA** system. Figure 7 shows an example of the kinds of questions DramaQA is designed to answer. Figure 8 shows a sample of DramaQA's annotated input data.

DramaQA selects from hand-picked multiple-choice options for each question, using a multi-layered approach that attempts to rank each answer four different ways, then selecting the answer with the highest summed score from all four methods. Overall, DramaQA was able to pick the correct answer 71.14% of the time, with higher accuracy for lower-difficulty questions and lower accuracy for higher-difficulty questions. Although DramaQA presents a very interesting system, it is limited by the need for heavy human involvement in the preparation of the data, including the need for extensive manual annotation and manual selection of candidate answers.

Figure 7: DramaQA Sample Questions

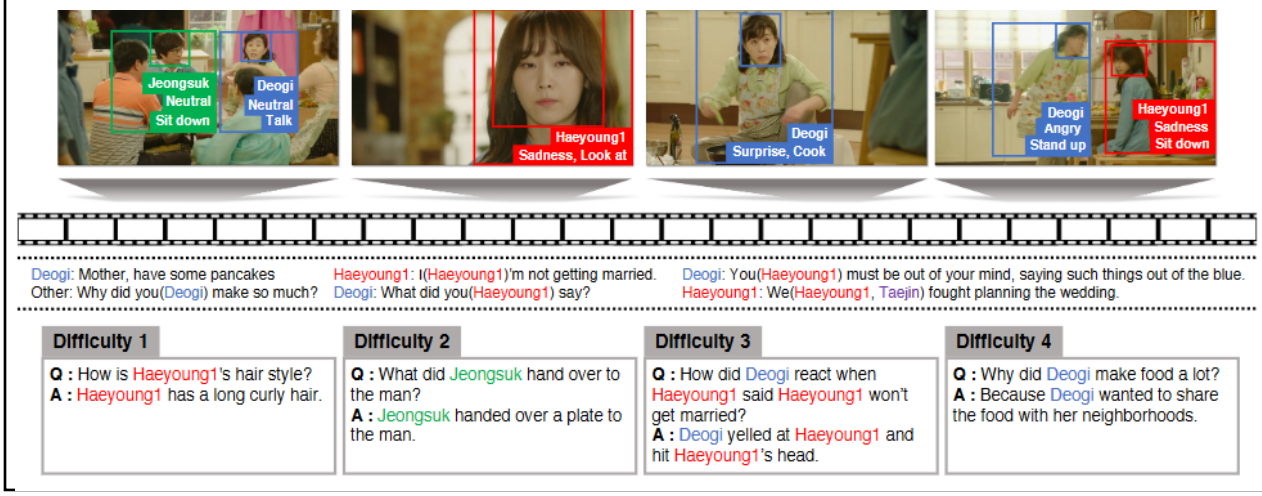
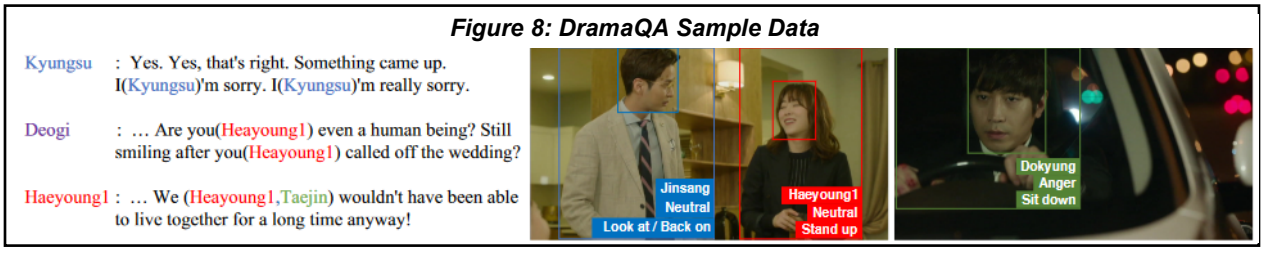


Figure 8: DramaQA Sample Data

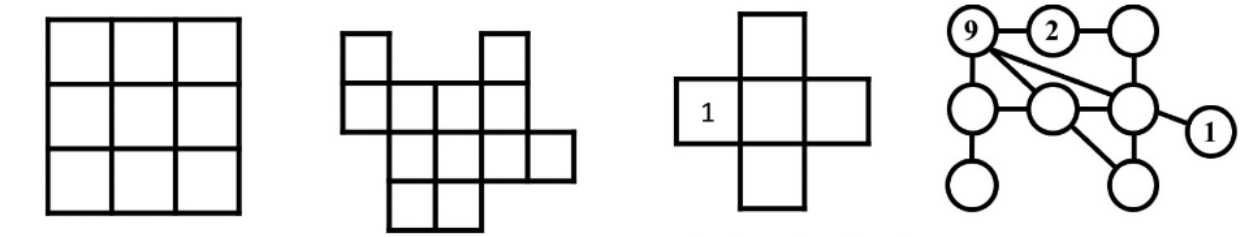


4.4 *A Prolog Application for Reasoning on Maths Puzzles with Diagrams*

Riccardo Buscaroli, Federico Chesani, Giulia Giuliani, Daniela Loreti, and Paola Mello. 2022. [9]

This complex system coordinates multiple AI capabilities with the goal of answering math puzzles intended for math competitions by elementary students. The system is designed to start with problems exactly as they would be presented to the children, input as an image file of a diagram and a prompt seeking a written answer. See Figure 9 below for several examples. The system identifies text regions on the image, identifies individual letters and words, parses the question, analyzes the diagram, classifies the problem into one of several classes of math puzzles, and then solves the puzzle.

Figure 9: Four Example Math Puzzles



The system is a response to the following challenge proposed by Chesani et al. (2017): *"By the middle of the 21st century, (a team of) fully autonomous agent(s) shall win a mathematical puzzle competition against primary school students, winners of the most recent competitions."* [11]

The code for this system has been made available as an open source code library, and a working demo of the system was also made available online at (http://games-ai.disi.unibo.it/games) but unfortunately the demo appears to be unavailable at time of writing.

This system was not groundbreaking in any one area of its operation. It relies heavily on already established open-source libraries to carry our most of its operations, including text identification, parsing, and machine vision. However, this work is unique is in the way it brings these components together and coordinates them into a system that is capable of much more than any of the subsystems could accomplish separately. Despite the recent successes of sub-symbolic techniques over logical systems in many areas of AI, a fully sub-symbolic system is ill-suited to solving problems which require explicit logical reasoning to arrive at the correct answer. The system presented here uses sub-symbolic components for parsing the text and analyzing the diagram but uses a logical puzzle reasoning library implemented in prolog as its final step of operation, bringing insights from sub-symbolic perception together into an explicitly logical system to generate the final correct answer.

4.5 *Question Sets and Challenges*

To encourage and facilitate further research into AI systems capable of answering short questions of the sort covered in this section, several large-scale question sets have been created and made available to researchers. This allows researchers to test different systems against the same question set and provides a ready pool of test data for new research. A few examples of question sets are explored below:

MCTest [44,59]: MCTest is *"a freely available set of stories and associated questions intended for research on the machine comprehension of text"* provided by Microsoft Research. The dataset was created with 500 short human-written stories, each with four associated multiple-choice questions. The dataset has grown to 660 stories at time of writing. The stories and questions are restricted to words and concepts that a 7-year-old would be expected to understand. *"By restricting the concept space, we gain the difficulty of being an open-domain problem, without the full complexity of the real world (for example, there will be no need for the machine to understand politics, technology, or to have any domain specific expertise)."*

bAbI [60]: bAbI is a project by Facebook AI Research, a branch of Meta Research. The project includes several question sets and challenges, including a set of prerequisite toy tasks for question-answering [52], six dialog tasks [3], the Children's Book Test [29], the Movie Dialog Dataset [3], the SimpleQuestions Dataset [4], and others.

Solving Mathematical Puzzles: A Challenging Competition for AI by Chesani et al. (2017) [11] puts forward the specific challenge that: *"By the middle of the 21st century, (a team of) fully autonomous agent(s) shall win a mathematical puzzle competition against primary school students, winners of the most recent competitions."* It provides several specific examples of the kinds of problems an AI should be able to solve, as well as describing prior works that have attempted to meet this challenge, and proposed methods for evaluating new systems.

4.6 *Question Answering Systems Discussion*

Question-Answering systems provide several interesting insights into AI story-understanding. One key insight is that several of these systems still rely on data that has been carefully hand-prepared by humans. *Choi et al*. (2021) relies on scripts and footage that have been heavily annotated, *Hill et al.* (2016) demonstrates a system that chooses between multiple-choice options that have been selected ahead of time, and *Bordes et al.* (2015) utilizes a knowledgebase where facts have been entered in a logical format as *(subject, relationship, object)*. This need for human preparation of data is a reflection of the fact that automatic parsers are still not able to extract the facts of a plain-text story with an acceptable level of accuracy, as was discussed in pervious sections. There would clearly be a benefit to a parser that was capable of automatically extracting logical declarations from a natural language story in plain text.

Neural networks have proven to be effective at answering simple questions from a given set of text. However, *Buscaroli et al.* (2022) show that this approach is insufficient for problems requiring complex reasoning. A multi-layered approach including both symbolic and sub-symbolic reasoning modules is required to effectively solve such problems.

Much of the prior research into QA systems has been motivated by a desire to improve search engine accuracy and is not geared towards building a more complete understanding of a story. The use of stories as source material for QA research is comparatively recent. In theory, a story model could be built by asking a series of questions about the story and then recording the answers in a structured way. However, I was not able to find any story modelling system that took this approach. Story modelling systems will be described in greater detail in the following section.

5. STORY MODELS

This section covers techniques and theories for modelling a story in its entirety, rather than just a single sentence or small snippet of text. I will be focusing on models that represent the specific entities and events that are described in the story, rather than models that aim to understand the overall narrative structure, themes, or other more abstract aspects of a story.

5.1 *Story Understanding Through Multi-Representation Model Construction*

Erik T. Mueller. 2003. [39]

Event Calculus is a logical notation that can be used to model of the events of a story. Mueller has written elsewhere about story modelling with event calculus [41] and the use of event calculus for **Common-Sense Reasoning** [40,42]. Event calculus is particularly useful for drawing logical inferences from a set of facts, especially when paired with a library of common-sense reasoning axioms. When reading a story, humans infer facts that are not explicitly stated within a text. Therefore, any AI system that aims to understand a story the way a human would, must be able use logical inference to create a more complete model of events than is available from what is explicitly stated in the text. This includes common-sense realizations such as “when a character drops an object the object will fall and continue to fall until it collides with the floor or another object,” and “when a character runs for a short period of time, their location changes.”

Four event calculus predicates are used to create this story model:

- *Happens(e, t) represents that an event e happens at time t.*
- *HoldsAt(f, t) represents that a fluent f holds at time t.*
- *Initiates(e, f, t) represents that if event e occurs at t then fluent f starts holding after t.*
- *Terminates(e, f, t) represents that if event e occurs at t then fluent f stops holding after t.*

Satisfiability solvers to can be used to determine facts that are implicit to the story. For example: to unlock a door a character must be awake, near the door, and the door must be in a locked state before it can be unlocked.. Thus, if the story states that a character has unlocked a door, we also know that all these other conditions are also true at that time. The Event calculus axiom for a character unlocking a door is represented as:

$$Happens(DoorUnlock(actor, door), time) \Rightarrow HoldsAt(Awake(actor), time) \wedge \neg HoldsAt(DoorUnlocked(door), time) \wedge HoldsAt(NearPortal(actor, door), time)$$

The primary gap in this model as a practical application for story understanding is that the initial encoding of explicitly stated facts within the story must be formulated as event calculus clauses by a human being, which prevents the system from forming a story model from plain text in a fully automatic way.

5.2 *Asking Hypothetical Questions About Stories Using QUEST*

Rachelyn Farrell, Scott Robertson, and Stephen G. Ware. 2016. [18]

QUEST [28] is a story model which was developed by cognitive scientists as a framework to predict how adults answer open-ended questions about finite sets of information. QUEST uses the **QUEST Knowledge Structure (QKS)** to represent short text narrative as a directed graph whose nodes are short sentence statements about the story. QUEST also provides a set of graph search procedures for estimating the **Goodness of Answer (GOA)** for questions about the story, which have been shown to match closely with the GOA scores given by human readers [28]. Prior works have used the QUEST GOA metric to validate computational models of narrative [10,14,45].

When humans make sense of a story, we consider not only events that actually occur within the story, but also compare these events to hypothetical situations that do not occur but could have occurred under different circumstances. For instance, a threat is understood as harm that could come to pass but hasn't yet, a hope or a goal is something a character would like to happen but hasn't happened yet. Human appreciation for stories is dependent on this ability to predict alternate narratives, things that could have happened but were avoided, or could still occur or not occur depending on the actions taken in the story. However, QKS is limited to reasoning about events that actually occur in a story and prior to this work had no mechanism for representing hypothetical scenarios.

To address this shortcoming of QKS, the authors present an expanded use of QUEST and QKS to represent hypothetical alternative paths that do not occur within a narrative but could have happened had things gone differently. QKS was used to generate a series of questions about what might have happened in a story if one or more elements of the story were changed. Multiple-choice answers were also provided for these questions. The new tool was validated by comparing the answers chosen by QKS to answers given by human readers to the same questions. This both proved the system's effectiveness and validated four hypotheses about the way human readers approach hypotheticals when thinking about a story. The four hypotheses, all confirmed by the study, are:

1) When presented with a hypothetical that makes critical events become impossible, people will answer using events from a story other than the one they read.
2) When presented with a hypothetical that makes new events possible that were not possible before, but that were part of a character's plan, people will also answer using events from a story other than the one they read.
3) When presented with a hypothetical about a critical event, people will answer using events from a story other than the one they read.
4) When presented with a hypothetical about a non-critical event, people will answer using events from the story they read.

5.3 *Narrative Models: Narratology Meets Artificial Intelligence*

Pablo Gervas, Birte Lönneker-Rodman, Jan Christoph Meister, and Federico Peinado. 2006. [20]

This paper provides an overview of narrative models in AI research and in the humanities, how theories from the two fields compare to each other, and how these different fields cooperate and inform one another. The authors are particularly concerned with the relative isolation of the two fields, noting that theories from the humanities are often ignored by AI researchers or are not deemed useful to NLP. When comparing theories used in these distinct but related fields, two key points emerge:

1. *Although narrative analysis and generation necessarily use different techniques in practice, they can share the abstract models underlying any theoretical and practical research on narrative;*
2. *Cooperation across the borders of our respective scientific disciplines shows that the challenges and obstacles encountered in both computational generation and computational analysis of narratives are closely related to conceptual key problems discussed in Narratology.* [20]

Work in this field can be broken down into two categories, characterized by the French words **histoire** meaning story or content, "what is told," and **discours**, meaning the text and presentation, "how it is told." One key difficulty is that narrative models in the humanities tend to be at least partially influenced by the history and culture the stories are situated in. Different cultures have different narrative structures, and readers view stories through the lens of their own cultural upbringing. This can make it difficult to extract universal patterns that apply across different times and cultures, which is often the goal for AI research. Another frustration in combining the humanities with AI research is that a great deal of NLP research deals with accurately parsing very short pieces of text, whereas literary analysis in the humanities often deal with texts that span thousands of pages. Most literary theories thus cannot be empirically tested with current AI techniques even when there is interest in doing so, because the texts under consideration are simply too large to be processed. There has been very little interest by AI researchers in studying narrative models which originated in the humanities.

This disconnect can be explained by the differing goals of humanities and AI researchers. AI research in NLP still struggles with basic issues of narrative understanding. Even representing or extracting the most straightforward, surface-level interpretation of the information contained in a sentence or paragraph remains a major challenge for AI systems. By contrast, literary analysis in the humanities is carried out by and for human beings who do not typically struggle to understand the surface-level meaning of text and are therefore not overly concerned with creating precise models of that information. Instead, literary analysis in the humanities focuses on more abstract concepts such as themes, tone, story structure, and commonalities and differences between stories and between cultures. Thus, models developed in one field are often neither useful nor interesting to researchers in the other.

5.4 *Story Models Discussion*

One of the main challenges with fully automated story modelling is the same issue that has been shown in all of these sections, the inability to parse the sentences of a plain-text natural language story into a series of logical statements. *Mueller* (2003) shows how event calculus can be used to extrapolate from a small set of facts that are directly stated in a story to build a whole model of assumed context derived from common-sense reasoning. However, to even begin this process the story first has to be broken down into a series of logical clauses by a human being. *Gervas et al.* (2003) note that one of the greatest difficulties in applying linguistic and narrative theories from the humanities to NLP applications is that theories in the humanities do not have to account for *"elementary communication issues."* Even seemingly structured systems such as QKS rely on basic reading comprehension skills on the part of the reader that NLP software cannot replicate.

To use the terminology presented in *Gervas et al.* (2003), I have focused on the *histoire* of a story rather than the *discours.* That is, I have focused on how an NLP software system can be made to understand the basic facts presented within a narrative rather than trying to understand the tone, theme, or overall narrative structure. There are systems which are capable of extracting some of these features from text even without a logical representation of the text's meaning. One good example is sentiment analysis, which is used to determine whether a short piece of text is generally positive or generally negative. For the sake of brevity, I have not included these sorts of systems in this review.

Also notably absent from this review is any story model that is focused on story generation rather than story understanding. While there is some overlap in the theories underlying either kind of system, story generators have a different set of objectives and challenges than story interpreters. There are numerous studies on story generation systems, which have also been excluded for the sake of brevity.

6. CONCLUSION

Natural Language Processing is a broad field of study covering a wide range of sub-topics. Solving complex problems in NLP may require drawing on insights from linguistics, humanities, and cognitive science as well as AI. This is almost certainly the case for the automatic parsing of stories. The ability to fully and accurately parse a story from plain-text natural language into a symbolic representation that can be logically manipulated and analyzed by a computer program could lead to a plethora of new capabilities. Several of the systems covered in this review have relied on such a representation as their starting point, but required that representation to be painstakingly encoded by human beings [12,39].

There is evidence to suggest that for complex problems such as this, a successful system may require both symbolic and sub-symbolic components working in coordination [9]. However, no system created to-date can successfully carry out the task of automatically parsing even the surface layer meaning of a story. New developments in neural networks have shown the ability to perform some reading comprehension tasks through the use of LSTM [29], and this may be the key to overcoming the ambiguity that prevents fully logical systems from parsing a substantial fraction of English sentences. If this capability could be more fully developed it could lead to many new possibilities within NLP. Since stories and narrative are so central to human understanding of the world, continuing to develop this capability in AI could be highly valuable to improving human-computer interaction with AI systems, and could lead to further insights into creating AI systems that reason in a more human-like way.

For now, however, AI understanding of stories remains out of reach. Current AI systems are only able to perform specific sub-tasks of story understanding, such as NER and question-answering tasks. Complex problems often require complex solutions, and *Buscaroli et al.* (2022) have shown that by breaking a problem down into smaller sub-problems and solving each of those individually, it is possible to create AI systems which can start with plain-text input, interpret that text into logical clauses, and perform logical analysis on those clauses. Their system was built to solve children's math puzzles, but it is not unreasonable to hope that in the future a similar approach could also be applied to the problem of story understanding.

7. BIBLIOGRAPHY

[1] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, Washington, USA, 1533–1544. Retrieved November 1, 2022 from https://aclanthology.org/D13-1160

[2] Kurt Bollacker, Robert Cook, and Patrick Tufts. 2007. Freebase: a shared database of structured general human knowledge. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2* (AAAI'07), AAAI Press, Vancouver, British Columbia, Canada, 1962–1963.

[3] Antoine Bordes, Y.-Lan Boureau, and Jason Weston. 2017. Learning End-to-End Goal-Oriented Dialog. DOI:https://doi.org/10.48550/arXiv.1605.07683

[4] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale Simple Question Answering with Memory Networks. *arXiv:1506.02075 [cs]* (June 2015). Retrieved April 13, 2022 from http://arxiv.org/abs/1506.02075

[5] Assma Boughoula, Aidan San, and ChengXiang Zhai. 2020. Leveraging Book Indexes for Automatic Extraction of Concepts in MOOCs. In *Proceedings of the Seventh ACM Conference on Learning @ Scale* (L@S '20), Association for Computing Machinery, New York, NY, USA, 381–384. DOI:https://doi.org/10.1145/3386527.3406749

[6] Brian Boyd. 2010. *On the Origin of Stories: Evolution, Cognition, and Fiction*. Harvard University Press.

[7] Jerome Bruner. 1991. The Narrative Construction of Reality. *Critical Inquiry* 18, 1 (October 1991), 1–21. DOI:https://doi.org/10.1086/448619

[8] Razvan Bunescu and Raymond Mooney. 2004. Collective Information Extraction with Relational Markov Networks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, 438–445. DOI:https://doi.org/10.3115/1218955.1219011

[9] Riccardo Buscaroli, Federico Chesani, Giulia Giuliani, Daniela Loreti, and Paola Mello. 2022. A Prolog application for reasoning on maths puzzles with diagrams. *Journal of Experimental & Theoretical Artificial Intelligence* 0, 0 (April 2022), 1–21. DOI:https://doi.org/10.1080/0952813X.2022.2062456

[10] Rogelio E Cardona-Rivera, Thomas W Price, David R Winer, and R Michael Young. Question Answering in the Context of Stories Generated by Computers. 19.

[11] Federico Chesani, Paola Mello, and Michela Milano. 2017. Solving Mathematical Puzzles: A Challenging Competition for AI. *AI Magazine* 38, 3 (October 2017), 83–96. DOI:https://doi.org/10.1609/aimag.v38i3.2736

[12] Seongho Choi, Kyoung-Woon On, Yu-Jung Heo, Ahjeong Seo, Youwon Jang, Minsu Lee, and Byoung-Tak Zhang. 2021. DramaQA: Character-Centered Video Story Understanding with Hierarchical QA. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 2 (May 2021), 1166–1174.

[13] Noam Chomsky. 1957. *Syntactic Structures*. De Gruyter Mouton. DOI:https://doi.org/10.1515/9783110218329

[14] David B Christian and R Michael Young. Comparing Cognitive and Computational Models of Narrative Structure. 6.

[15] Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-Semantic Parsing. *Computational Linguistics* 40, 1 (March 2014), 9–56. DOI:https://doi.org/10.1162/COLI_a_00163

[16] Ernesto D'Avanzo, Alfio Gliozzo, and Carlo Strapparava. Automatic Acquisition of Domain Information for Lexical Concepts. 7.

[17] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Edinburgh, Scotland, UK., 1535–1545. Retrieved November 1, 2022 from https://aclanthology.org/D11-1142

[18] Rachelyn Farrell, Scott Robertson, and Stephen G. Ware. 2016. Asking Hypothetical Questions About Stories Using QUEST. In *Interactive Storytelling* (Lecture Notes in Computer Science), Springer International Publishing, Cham, 136–146. DOI:https://doi.org/10.1007/978-3-319-48279-8_12

[19] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Association for Computational Linguistics, Ann Arbor, Michigan, 363–370. DOI:https://doi.org/10.3115/1219840.1219885

[20] Pablo Gervas, Birte Lönneker-Rodman, Jan Christoph Meister, and Federico Peinado. 2006. Narrative Models: Narratology Meets Artificial Intelligence. In *Narrative models: Narratology meets artificial intelligence*, Genoa, Italy, 44–51.

[21] Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics* 28, 3 (2002), 44.

[22] Alfio Gliozzo. 2006. Semantic Domains and Linguistic Theory. In *Narrative models: Narratology meets artificial intelligence*, Genoa, Italy, 6.

[23] Alfio Gliozzo, Claudio Giuliano, and Carlo Strapparava. 2005. Domain Kernels for Word Sense Disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational*

Linguistics (ACL'05), Association for Computational Linguistics, Ann Arbor, Michigan, 403–410. DOI:https://doi.org/10.3115/1219840.1219890

[24] Alfio Massimiliano Gliozzo. 2006. The GOD model. In *Demonstrations*, 147–150. Retrieved October 31, 2022 from https://aclanthology.org/E06-2016

[25] Alfio Gliozzo and Carlo Strapparava. 2005. Domain Kernels for Text Categorization. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, Association for Computational Linguistics, Ann Arbor, Michigan, 56–63. Retrieved October 31, 2022 from https://aclanthology.org/W05-0608

[26] Alfio Gliozzo and Carlo Strapparava. 2005. Cross Language Text Categorization by Acquiring Multilingual Domain Models from Comparable Corpora. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Association for Computational Linguistics, Ann Arbor, Michigan, 9–16. Retrieved October 31, 2022 from https://aclanthology.org/W05-0802

[27] Alfio Gliozzo, Carlo Strapparava, and Ido Dagan. 2004. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech & Language* 18, 3 (July 2004), 275–299. DOI:https://doi.org/10.1016/j.csl.2004.05.006

[28] Arthur C. Graesser, Sallie E. Gordon, and Lawrence E. Brainerd. 1992. QUEST: A model of question answering. *Computers & Mathematics with Applications* 23, 6–9 (March 1992), 733–745. DOI:https://doi.org/10.1016/0898-1221(92)90132-2

[29] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv:1511.02301 [cs]* (April 2016). Retrieved April 12, 2022 from http://arxiv.org/abs/1511.02301

[30] Christopher R. Johnson, Charles J. Fillmore, Esther J. Wood, Margaret Urban, Miriam R. L. Petruck, Collin F. Baker, Charles J. Fillmore, and et al. 2001. The FrameNet Project: tools for lexicon building.

[31] Vijay Krishnan and Christopher D. Manning. 2006. An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Sydney, Australia, 1121–1128. DOI:https://doi.org/10.3115/1220175.1220316

[32] John Lafferty, Andrew McCallum, and Fernando C N Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proc. ICML* (June 2001), 10.

[33] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. *Proceedings of NAACL 2016* (April 2016). Retrieved April 28, 2022 from http://arxiv.org/abs/1603.01360

[34] Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4* (AS '02), Association for Computational Linguistics, USA, 45–51. DOI:https://doi.org/10.3115/1118162.1118168

[35] John Lyons. 1977. *Semantics: Volume 2*. Cambridge University Press.

[36] Bernardo Magnini and Gabriela Cavaglia. 2000. Integrating Subject Field Codes into WordNet. *LREC* 1413, (2000), 6.

[37] Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2001. Using Domain Information for Word Sense Disambiguation. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, Association for Computational Linguistics, Toulouse, France, 111–114. Retrieved October 31, 2022 from https://aclanthology.org/S01-1027

[38] George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the Acm* 38, (1995), 39–41.

[39] Erik T. Mueller. 2003. Story understanding through multi-representation model construction. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning - Volume 9* (HLT-NAACL-TEXTMEANING '03), Association for Computational Linguistics, USA, 46–53. DOI:https://doi.org/10.3115/1119239.1119246

[40] Erik T. Mueller. 2004. Event calculus reasoning through satisfiability. *Journal of Logic and Computation* 14, (2004), 2004.

[41] Erik T. Mueller. 2004. Understanding script-based stories using commonsense reasoning. *Cognitive Systems Research* 5, 4 (December 2004), 307–340. DOI:https://doi.org/10.1016/j.cogsys.2004.06.001

[42] Erik T. Mueller. 2014. *Commonsense reasoning: an event calculus based approach*. Morgan Kaufmann.

[43] Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning - CoNLL '09*, Association for Computational Linguistics, Boulder, Colorado, 147. DOI:https://doi.org/10.3115/1596374.1596399

[44] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, Washington, USA, 193–203. Retrieved April 27, 2022 from https://aclanthology.org/D13-1020

[45] M. O. Riedl and R. M. Young. 2010. Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research* 39, (September 2010), 217–268. DOI:https://doi.org/10.1613/jair.2989

[46] Stuart Russel and Peter Norvig. 2010. *Artificial intelligence: A Modern Approach* (Third Edition ed.). Prentice Hall, Upper Saddle River, NJ.

[47] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. Retrieved October 29, 2022 from http://arxiv.org/abs/cs/0306050

[48] Fei Sha and Fernando Pereira. 2003. Shallow Parsing with Conditional Random Fields. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 213–220. Retrieved October 29, 2022 from https://aclanthology.org/N03-1028

[49] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems*, Curran Associates, Inc. Retrieved April 13, 2022 from https://proceedings.neurips.cc/paper/2015/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html

[50] Robert D Van Valin. 1993. ROLE AND REFERENCE GRAMMAR. *Work Papers of the Summer Institute of Linguistics* 37, (1993), 12.

[51] Robert D. Van Valin jr. 1993. Advances in Role and Reference Grammar. *Advances in Role and Reference Grammar* (1993), 1–583.

[52] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv:1502.05698 [cs, stat]* (December 2015). Retrieved April 26, 2022 from http://arxiv.org/abs/1502.05698

[53] Ludwig Wittgenstein. 19665. *Philosophical Investigations*. The Macmillan Company, New York.

[54] Yong Zhang and Weidong Xiao. 2018. Keyphrase Generation Based on Deep Seq2seq Model. *IEEE Access* 6, (2018), 46047–46057. DOI:https://doi.org/10.1109/ACCESS.2018.2865589

[55] Definition of syntax | Dictionary.com. *www.dictionary.com*. Retrieved October 26, 2022 from https://www.dictionary.com/browse/syntax

[56] *Overview of MUC-7/MET-2*. Retrieved October 29, 2022 from https://apps.dtic.mil/sti/citations/ADA633464

[57] Home. *Einstein Bros. Bagels*. Retrieved November 17, 2022 from https://www.einsteinbros.com/

[58] Project Gutenberg. *Project Gutenberg*. Retrieved April 26, 2022 from https://www.gutenberg.org/

[59] Machine Comprehension Test (MCTest). Retrieved November 2, 2022 from https://mattr1.github.io/mctest/

[60] bAbI - Meta Research. *Meta Research*. Retrieved April 26, 2022 from https://research.facebook.com/downloads/babi/