An Approach To The Winograd Schema Challenge Based On Semantic Classification Of Events And Adjectives

by

Turash Mosharraf

B.Sc., Bangladesh University of Engineering and Technology, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2019

Approval

Name:	**Turash Mosharraf**
Degree:	**Master of Science (Computing Science)**
Title:	**An Approach To The Winograd Schema Challenge Based On Semantic Classification Of Events And Adjectives**
Examining Committee:	**Chair:** Eugenia Ternovska Associate Professor
	James Delgrande Senior Supervisor Professor
	Fred Popowich Supervisor Professor
	Angel Chang External Examiner Assistant Professor
Date Defended:	**November 25, 2019**

Abstract

The Winograd Schema Challenge is an interesting problem that can be considered as an alternative to the renowned Turing test for measuring the intelligence of artificial agents. Originally proposed by Hector Levesque in 2011, the challenge gained significant popularity among the researchers of Natural Language Understanding (NLU). It is a special class of coreference resolution problem and the challenge involves finding the correct referent for a given pronoun or possessive adjective in a short text. The Winograd Schema was proposed to encourage human-like reasoning and therefore the problems were carefully designed so that the statistical methods which solely depend on lexical correctness and word associativity without any logical reasoning will not be able to solve these problems reliably. The designers encouraged the use of commonsense based logical methods to solve the Winograd Schema. However, at present, in terms of accuracy, the most advanced logical methods are far from human-like accuracy and also significantly outperformed by the statistical methods. The major challenge for the logical methods is the difficulty of finding a general model for the Winograd Schema. The objective of this research is to introduce a commonsense based logical method that can achieve competitive accuracy with the statistical methods on a specific form of the Winograd Schema problems. Our approach is based on semantic classification of common verbs and adjectives, creating a commonsense knowledge base, and using the knowledge base to align the pronoun with the correct referent. We have evaluated the performance of our approach and compared the accuracy with the state of the art statistical approach on two benchmark problem sets: $WSCL$ and $WSCR$.

Keywords: Winograd Schema; Common Sense Reasoning; Natural Language Understanding; Sentiment Analysis

Dedication

To My Parents And Sister.

Acknowledgements

I want to thank my senior supervisor for his guidance. I am also thankful to my family and friends for their support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the flourishing of artificial intelligence, the question of whether machines are capable of displaying human-like intelligence is getting more and more attention in the research community. For a long time, the renowned Turing test proposed by Allan Turing has been considered to be the most reliable method to judge an artificial intelligent (AI) agent. The Turing test involves a human evaluator who tries to distinguish between a human and an AI agent by natural language conversation only. If the evaluator fails to distinguish, the AI agent is considered to pass the test. However, Hector Levesque [19] pointed out several ways in which the AI agent can trick the evaluator. One of the ways is to use exclamatory or emotional words that are frequent in human conversation. For example, "This is a beautiful day" can be changed to "Wow! This is such a beautiful day!" to make it sound more human-like. Even though adding words like "wow" or "such" adds no new information or reasoning, they are frequently used to express emotion in human speech and using them may be enough to bias the evaluator's judgement. Therefore, the Turing test is not the best measurement of human-like intelligence.

As an alternative, Levesque proposed using small snippets of texts involving two parties and a pronoun along with a question to determine the correct reference for the pronoun. The correct referent is guaranteed to be one of the parties and therefore an agent can not hide its identity by adopting any elusive technique. Compared to the Turing test, Levesque's method can serve as a better measure of how much an agent understands natural language and use human-like reasoning to make decisions. Levesque named this test after Terry Winograd [56] and since then, it is known as the Winograd Schema Challenge.

1.1 What Is The Winograd Schema?

The following text proposed by Winograd serves as the first example of the Winograd Schema:

- The city councilmen refused the demonstrators a permit because they feared violence. Who feared violence?

- The city councilmen refused the demonstrators a permit because they advocated violence. Who advocated violence?

The sentences are identical except for a single place where "feared" is replaced by "advocated". However, the answers are different. The answer to the first question is "the city councilmen" whereas the answer to the second question is "the demonstrators".

The problem is simple enough for humans. Anyone fluent in English, even a child, can answer correctly without much thinking. However, the question requires reasoning with commonsense knowledge which is not trivial for an artificial agent. Interestingly, the accuracy of even the most sophisticated natural language processing (NLP) program is close to a random guess for this problem. Therefore, this test points out one of the most fundamental differences between human level and machine level understanding.

In chapter 2, we discuss and compare the performances of the most advanced AI programs currently available and present their strengths and weaknesses. In general, the greatest weakness of AI programs is the failure to apply appropriate commonsense knowledge. Moreover, humans can use their eyes, ears and other sensory organs to collect information which help them to accumulate a rich and versatile knowledge base. The ability of a human brain to match new knowledge with the old and determine its applicability based on a given context is far superior to a machine that often fails to find relevant knowledge.

1.2 Why Solving the Winograd Schema is Difficult

The Winograd Schema is specially designed to force an agent to use commonsense reasoning on a semantic and pragmatic level. Current NLP solutions based on deep learning use lexical features and word associativity, which lacks semantic and pragmatic information. These methods work effectively for tasks which are based on grammatical correctness (i.e., machine translation, sentence completion, text generation). However, for the Winograd Schema, the incorrect answer is always grammatically correct.

Erik Cambria [4] analyzed the progress of NLP research and divided it into three major eras: i) Syntactic, ii) Semantic, iii) Pragmatic. He stated that current research is transitioning from the syntactic era to the semantic era and it might be around the year 2050 before we enter the pragmatic era. Most of the Winograd Schema problems require semantic reasoning.

The following problem can be considered:

The man could not lift his son because he was too weak. Who was too weak?

The correct answer is obviously "the man". However, "the son", which is the incorrect answer, is syntactically valid as well. Both of the candidates are male and singular in number. It is required to reason about the subject of "lift" and the subject of "weak" in order to

determine the correct answer. This reasoning involves semantic analysis of the word “lift” and “weak”.

Although the semantic analysis is challenging, pragmatic analysis is much more difficult for current AI programs. Some Winograd Schema problems are presented in a way that makes both the alternatives semantically valid. The following example illustrates this:

> Tom threw his schoolbag down to Ray after he reached the top of the stairs. Who reached the top?

The correct answer is “Tom”. However, “Ray”, which is the incorrect answer, is syntactically and semantically valid as well. Even if the semantics of “top”, “stair”, “throw down” is understood, it is valid to imagine a world where Tom was standing on a higher platform over the stairs (e.g., the balcony or the roof) and threw the bag down after Ray reached the top of the stairs. Pragmatic thinking would rule out this interpretation as “far fetc.hed” because there was no mention of any higher platform in the sentence itself. Current NLP research is still far from such pragmatic reasoning.

1.3 Current Trends and Progress

In the field of Natural Language Processing, statistical and probabilistic approaches are very successful. These methods are capable of capturing syntactic features from texts and are widely used in tasks like text generation, text completion, machine translation, and language modelling. The Winograd Schema problems can be represented as a special form of text completion by replacing the pronoun by a blank and predicting the appropriate referent to fill the gap. However, each Winograd Schema problem contains a special word which, if replaced by an alternative word, switches the answer. The probability for both referents are calculated from the whole context and therefore statistical methods fail to switch answer if a single word is replaced. Because of this property, it is hard to be solved by statistical approaches.

To overcome this limitation, recent works [17] [15] [43] focus on transfer learning where the language model learned on a large text corpus is tuned on a set of sentences with the same structure as the Winograd Schema. The tuning allows the model to learn about the special word property and significantly improves the performance. These approaches can achieve almost human-like accuracy (90% for [43] compared to 94.1% for humans). However, the authors also acknowledged that this high accuracy is a result of an inherent bias in the dataset instead of actual understanding. Moreover, this approach still learns only word associativity and not the semantics and, therefore, can not beat the human accuracy in an unbiased dataset. It is evident that commonsense and logical reasoning is necessary in order to achieve human-like competency on the Winograd Schema. In theory, the commonsense based methods are more suitable for the Winograd Schema Challenge.

Unfortunately, in practice, the performance of logical methods is not impressive. Because of the diversity of commonsense knowledge and reasoning types, there is no general logical method for the Winograd Schema. Even the most advanced logical approaches [46], [12] are significantly outperformed by the current state of the art statistical approach (57.1% for [12] compared to 90% for [43]) because of their incapability to incorporate different types of logical reasoning that are necessary to solve the Winograd Schema problems. At present, all logical approaches target specific forms of the Winograd Schema and can not generalize well beyond the form.

The structure of the Winograd Schema ensures the presence of two candidate referents and a pronoun or possessive adjective. The clauses of the candidates and the pronouns usually denote two distinct events and are often connected by discourse connectives. Based on the relationships of the events and the connectives, the problem set can be categorized into different subsets where each subset requires different forms of reasoning (e.g., causal, temporal, epistemic). Most of these subsets are limited enough to be solved using semantic analysis without the use of reasoning about pragmatics. There are many works that target specific subsets and provide automated solutions to solve them. There exists a logical framework described by Schuller et al. [45] to solve the most difficult pragmatic subset. However, there is no automated solution based on the framework yet. In this thesis, we limit our analysis within the semantic level and develop an approach to solve a subset of the Winograd Schema using semantic analysis. We assume that, each words in the sentence is meaningful and our approach uses the meanings and relationships of the lexicons to solve the Winograd Schema. Therefore, in this thesis, we only use lexical semantics.

1.4 Our Approach: A Different Path

In order to solve the Winograd Schema, it is crucial to identify the connection between the pronoun and the candidate referents. Since both refer to the same entity, they must share similar semantic features. One of the main reasons behind the success of humans to solve the Winograd Schema is the ability to detect the connection between the pronoun and the candidate referents based on their common semantic features. One of the most common semantic features is sentiment polarity, which is a sense of positivity, negativity, or neutrality towards an entity. Although, in some cases, sentiment polarity can be argued to be subjective, there are a lot of words that are almost universally negative or positive. For example, “bad”, “weak” and “corrupt” are always negative compared to “good”, “strong”, and “honest”. Sentiment polarity toward each of the entities mentioned in the text plays an important role in associating a pronoun to its referent. Based on this observation, we create a knowledge base of common verbs and adjectives and their polarities.

Sentiment polarity of words has been used for a long time in opinion mining and sentiment analysis. However, it is typically used as a single atomic feature with only three possible

values: positive, negative, or neutral. Tuvery et al. [51] proposed a multidimensional vector model of polarity. According to Tuvery, the features are: "emotion", "moral/ethic", "character", "weather", "color", "quantity", "appearance", "material", "shape", "touch", "taste", "dimension", "chronologic", and "geographic". This model of polarity contains significantly more information compared to the traditional model and therefore we adopt this model in our approach. We observe that some of the features mentioned by Tuvery et al. depend on others. For example, "moral/ethic" and "character" are correlated. For classification, it is important to have a feature set with minimum interdependency. Therefore, we organize the features of Tuvery et al. into a hierarchical relationship. In each hierarchy level, we have a set of independent features of polarity.

At the topmost level, we have only one feature: "polarity", same as the traditional concept. This denotes a general notion of positivity, negativity or neutrality. For example, "bad" is always considered to have a negative value compared to "good", and "strong" is always considered to have a positive value compared to "weak". Our objective is to classify common verbs and adjectives to be used for solving the Winograd Schema. However, we found examples of adjectives that are harder to classify using this type of polarity. A notable example is "aggressive", which can be associated with "brave" (a positive adjective), or "arrogant" (a negative adjective). To have a better classification metric, we need to go to the lower levels in the hierarchical structure.

In the second level, we have "ability" and "moral/ethic" as two independent features of the polarity. "Ability" is a generic feature that denotes the combination of all other features of Tuvery except "moral/ethics". If we ignore the "neutral" value, both of the features can have 2 different values and, therefore, any word can be classified into 4 disjoint classes. Although the number of classes increases from 2 to 4, using second level features of polarity leads to effective classification. If we consider our example of "aggressive", "arrogant", and "brave", we observe that "arrogant" has negative ethical value and "brave" has positive ability. Therefore, "aggressive" can have negative "moral/ethic" and positive "ability".

The next level of polarity hierarchy would break down "ability" into two more independent features. The further down we go into the hierarchy, the better classification can be obtained. However, the number of classes increase exponentially with the number of features. If we allow a word to have "positive" and "negative" value for each of the features, for N number of features, we have 2^N classes. In this thesis, we work with only the topmost two levels. One of our objectives is to observe how much accuracy can be gained by going from level 1 to level 2.

All the words are stored in the knowledge base based on their classification. However, Peng et al.[37] argued that the polarity of a word depends on the context in which the word is used. Therefore, our knowledge base can only store prior probabilities which needs to be tuned to be used for the Winograd Schema problems. When we analyze a problem,

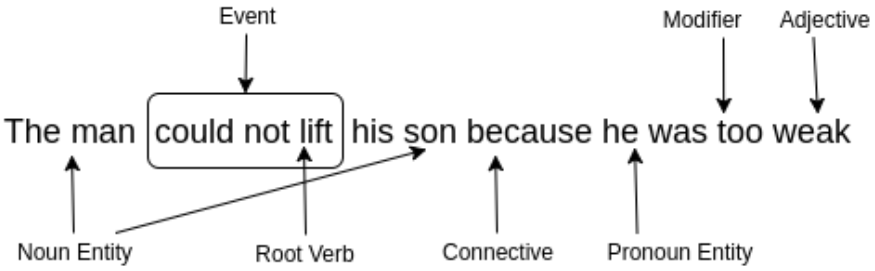


Figure 1.1: Events, Entities, Verbs and Adjectives from "The man could not lift his son because he was too weak"

we search the knowledge base for the polarities of all the verbs and adjectives used in the Winograd Schema and tune them according to the context using a set of rules.

1.4.1 Events and Entities

Events and Entities are the building blocks of the Winograd Schema problems. After determining the polarities of all the verbs and adjectives, we use them to determine the polarities of all the events and entities. The human cognitive process analyzes texts as chains of events [55]. The events may be related or unrelated. Since the Winograd Schema are carefully designed to be easily solvable by commonsense and logic, the chain of events is always related. Events may contain one or more words and are often associated with subjects and objects. Events must contain at least one verb, which acts as the root and may be associated with modifiers or complimentary words. For example, in the text: "The man could not lift the son", "can not lift" is the event where "lift" acts as the root and "can not" acts as the modifier. "the man" and "the son" are the entities that are associated with the event.

The polarity of an event is obtained by the polarity of the root verb, which is tuned by the modifier. All the entities are either nouns or pronouns and are associated with at least one event. The relation between an event and an entity can be very diverse. However, "subject" and "object" are the most useful relationships which determine the polarity of the entities. In the above example, "the man" and "the son" are the subject and object, respectively.

The adjectives also affect the polarity of the entities because they act as the traits associated with the entities. Events and entities are illustrated in Figure 1.1.

1.4.2 Classification of Verbs and Adjectives from Events and Entities

One of the most important steps of our method is to create the knowledge base. We implement a system that starts with a small set of manually classified adjectives. As the initial set, we use the subjectivity lexicons by Wilson [55]. Once we form the initial knowledge base, we expand it automatically. In order to expand the knowledge base, we implement an automatic system which accumulates more verbs and adjectives by performing logical infer-

ence using a set of large text corpora. The relation between an adjective and its associated entity is straightforward. An adjective like "generous", which has a positive polarity value, would ensure the associated entity to have positive polarity value as well, unless negated or discounted by a modifier like "not" or "less". Therefore, the polarity of an adjective is same as the polarity of its associated entity and either of them can be determined directly from the other.

The classification of verbs is performed based on its event and the entities associated with the event. Each event has a subject and, optionally, an object. Moreover, events are capable of modifying the polarity of a subject or object. For example, in the sentence "John's skill is improved", the event "improve" changes the polarity of the subject "John" from negative to positive. Therefore verbs are classified based on the prior and posterior condition of the subject and object of the event. Chapter 3 explains the classification process in detail.

1.4.3 Solving the Winograd Schema

To solve a Winograd Schema, we first parse the text using a semantic parser. We use the parser to create events. If the event contains multiple words, we use a set of rules to determine the polarity of the event from the constituent words. All the entities and their subject-object relationship to the events are also extracted. Finally, we use our logical inference module which determines the polarities of the entities from the polarities of their adjacent events and relationships and use it to align the pronoun to its referent. All of the steps are discussed in detail in chapter 4.

1.5 Structure of the Thesis

The rest of the thesis is structured as follows: In chapter 2, we present the background of the Winograd Schema challenge, different approaches, and their pros and cons. We also compare their performances with actual human performances. In chapter 3, we discuss in detail how we arrange the polarity features in a hierarchical structure and how we use the structure to create a knowledge base. Chapter 4 explains our method to solve the Winograd Schema problems using our knowledge base. Moreover, this chapter also presents the experimental results and compares the performance of our approach with the previous ones and the state of the art. In chapter 5, we discuss the contribution and limitation of this thesis as well as possible future works.

Chapter 2

Related Work

In this chapter, we discuss previous approaches to the Winograd Schema problems along with their advantages and drawbacks. A brief comparison of their results to real human participants is presented afterwards. Most of the works can be classified into two major categories: statistical and logical. However, because of the limitations of standalone statistical or logical approaches, some approaches try to combine both of them in order to learn the underlying reasoning necessary to solve this challenging problem. We will discuss in detail how the underlying ideas used to build these systems motivate the design of our system. Additionally, we discuss a recent study showing the performance of real humans on the Winograd Schema problems and compare their performance with intelligent systems.

2.1 The Winograd Schema Challenge

The Winograd Schema Challenge was first proposed by Levesque et al.[19] as an alternative to the renowned Turing Test. The schema gains its name from Terry Winograd [56]. Levesque prepared an initial set of such problems which was later enhanced and refined by L. Morgenstern [30],[31] and E. Davis [8].

Levesque proposed the Winograd Schema problems as small English texts involving a single question. The following example illustrates:

"The trophy did not fit in the brown suitcase because it was too small. What was too small?"

- Answer 0: The trophy
- Answer 1: The suitcase

The correct answer is: "The suitcase". The answer is obvious by any untrained human.

2.1.1 Structure Of The Winograd Schema

The Winograd Schema problems are designed with the following structure:

- Two parties are mentioned in a sentence by noun phrases.
- A pronoun or possessive adjective is used in reference to one of the parties. However, it must agree with the number and gender and be grammatically compatible with both parties.
- The question involves determining the reference. In some versions, the questions are omitted.
- There is a special word in the sentence. If the special word is replaced by an alternative word, the answer changes. However, the alternative sentence remains syntactically and semantically valid.

Levesque stated that, in the ideal case, the vocabulary of the Winograd Schema should be restricted enough that even a child can answer correctly.

The presence of the special word and the alternative word ensures that no program can solve these problems without logical reasoning. Clever programs which use lexical features like word ordering fails to change the answer when special words are replaced by an alternate word because everything else remains exactly in the same order. Therefore, it makes the Winograd Schema "Google-Proof" and using a large text corpus would not help much to answer the question.

Background knowledge is necessary to solve the Winograd Schema. However, the knowledge is not present in the sentence itself.

2.1.2 Features Of The Winograd Schema

Levesque identified three features that a Winograd Schema must have. A Winograd Schema must be:

- Easily solvable by untrained humans.
- Not solvable by simple methods like selectional restriction.
- Google-Proof: Not solvable by searching through a large text corpora and learning word associativity.

2.1.3 Designing The Winograd Schema

Designing a Winograd Schema is difficult and requires careful analysis. There are some common pitfalls which can violate some of the features.

- Not Google-Proof: The following example illustrates the pitfall:

  "The women stopped taking the pills because they were pregnant/carcinogenic. Who were pregnant/carcinogenic?"

A clever program can easily find the associativity of "women" and "pregnant" or "pills" and "carcinogenic" by searching a large text corpus. Therefore, this example is not google-proof.

- Over ambiguity: Solution to a problem of the Winograd Schema should be obvious enough to humans. It is possible to design problem instances which are ambiguous even to humans. The following example is one of them:

  "Frank was pleased when Bill revealed that he was the winner of the competition. Who was pleased?"

  The correct answer is Frank because he was pleased. The problem is that, it is not unreasonable for Frank to be pleased of Bill's win either. Therefore, the question is too difficult to be used as a Winograd Schema problem. One improvement would be to add more information like "Bill is a rival of Frank" which would give Frank more reason not to be pleased at Bill's win.

2.2 Solutions To The Winograd Schema: Statistical Approaches

The accuracy and efficiency of machine learning based approaches on classical natural language processing problems like machine translation and sentence completion inspired many researchers to consider statistical and probabilistic approaches for solving the Winograd Schema problems. Most of the early works on the Winograd Schema focus on learning lexical and semantic features of the sentences using state of the art machine learning approaches to determine the reference of the pronoun. These methods work reasonably well on the Pronoun Disambiguation Problem (PDP) [7], which has some similarities to the Winograd Schema. Given two events and their associated entities along with their semantic roles, these methods try to find the probability distribution of the events that appear in the same sentence.

2.2.1 Rahman's Approach

Rahman et al. [39] created their own set of hard coreference problem similar to Levesque's set. In their set, each sentence contains two clauses that are usually connected by a discourse connective(e.g., "because", "and", "although", etc.). One of the clause contains the candidate mentions and the other contains the target pronoun. The target pronoun agrees with the number and gender of both of the candidates, thereby making the problem set similar to Levesque's set in some aspects.

The data set contains 941 sentence pairs (1882 sentences). Among them, 70% (1318 sentences) are used to train the classifier model and the remaining 30% (564 sentences) are used to test the accuracy. The authors claimed that their approach achieves 73.05% accuracy on the test set.

Component	Number of Features
Narrative chains	1
Google	4
FrameNet	4
Heuristic Polarity	4
Learned Polarity	4
Connective Based Relation	1
Semantic compatibility	3
Lexical Features	68,331
Total	68,350

Table 2.1: The features used in Rahman et al's approach

The features used by Rahman et al. are summarized in Table 2.1. The features are described below:

- Narrative Chains: This feature considers the chain of events and roles which an entity may go through. The events are organized in a causal relationship so that the events and roles which appear early in the chain are the cause of the events and roles that appear later. Chambers et al. [5] developed narrative chains of different lengths. One example of narrative chain is: "borrow-s invest-s spend-s pay-s raise-s lend-s". This suggests that, playing a subject role in a "borrow" event may be a plausible cause of playing a subject role in "invest" event.

- Google: These features represents the response counts from Google for automatically generated queries. Consider the sentence: "Lions eat zebras because they are predators". replacing the pronoun ("they") with the candidates ("lions" or "zebras") results in two possible clauses: "Lions are predators" and "Zebras are predators". A Google query with "Lions are predators" is likely to generate many more responses than a query with "Zebras are predators".

- FrameNet: The feature is similar to Google features except that, proper nouns are replaced by their roles. In the sentence, "John killed Jim, so he was arrested", John and Jim both participate in the "kill" event with John as "killer" and Jim as "victim". replacing their role in original sentence, the queries become: "killer was arrested" and "victim was arrested".

- Heuristic Polarity: When analyzing a sentence, humans cognitively imposes some degree of positivity or negativity to all the entities involved. This is an automatic process in human brains to judge other entities according to their perception of the world. For example, in the above example where John killed Jim, a human listener would subconsciously assign a negative feeling towards John, who was playing the subject

role in a "kill" event. However, if the sentence was modified to be "John helped Jim", the feeling toward John would be a positive one. Wilson el al. [55] proposed a method to extract such contextual polarities from a sentence and created a list of words with their prior polarities. In this feature, the prior polarity is modified in a given context using some heuristic rules.

  - The polarity of a subject of an event is same as the event whereas the polarity of an object is the reversed.
  - If there is a negative modifier attached to the event (e.g., less), then the polarity is reversed.
  - if the clause in which the event occurs is connected by a negative discourse connector, the polarity of all entities in the clause is reversed.

  The contextual polarities of a pronoun should match the candidate with similar contextual polarity

- Learned Polarity: This feature is similar to the heuristic polarity except that no heuristic rules are applied to determine the contextual polarity from prior polarity. Instead, the polarities of all entities are determine by a machine learning automated tool: "OpinionFinder" [54].

- Connective Based Relation: This feature is used to determine the probability of two entities who play different roles in two events to be connected by a discourse connective. It can determine probability of a subject of "bully" to be an object of "punish", given that the connective is "so". This feature is useful because changing the connective from "so" to "but" may result in changing the correct answer.

- Semantic Compatibility: Sometimes, two words which often appear together in a sentence are not semantically compatible to participate in an event-entity relationship. For example: aeroplanes are made of "aluminium" and are likely to participate in "fly" events and therefore any query with "aluminium" and "fly" would result in a lot of response from google. However, "aluminium" as a metal can not be subject to "fly" event. this feature acts as a filtering to other features which depends solely on the number of query responses.

- Lexical Features: In addition to the above mentioned features, Rahman et al. used lexical features based on structure of the sentences. This set of features are not very suitable for the Winograd Schema. However, they may work relatively better in a problem set that contains less ambiguous sentences.

Rahman's approach uses machine learning to learn the weights of the features. Although this approach uses a large number of features, the features may not have enough impact

if the training set is not rich enough. Moreover, the features are very sparse and some of them have non-zero values only on rare cases. However, the approach remains as a useful baseline for many of the later approaches.

2.2.2 Peng's Approach

Peng et al. [37] addressed some drawbacks of Rahman's approach and proposed an alternative approach to overcome those limitations. The authors pointed out that injecting the semantic properties as features may lose their impact because of their sparsity. The authors argued that solving an Integer Programming Problem (ILP) using the semantic properties as constraints would result in higher impact. They formalized their concept of Predicate Schema, studied two types of predicate schema and provided a method to compile the predicate schema into constraints of an ILP. They categorized all pronoun resolution problems into two categories:

- Easy: Where the candidates and pronouns differ in number and gender.
- Hard: Where the candidates and pronouns agree with all structural attributes, (e.g., The Winograd Schema, Rahman's problem set). These problems require background knowledge and common sense.

Peng's approach outperforms Rahman's approach by achieving 76.76% accuracy on hard problems.

The authors describe Predicate Schema as a formalism to determine the probability of finding a pattern of words in a large corpus. Peng et al. describe two types of Predicate Schema:

- Type 1: Given a predicate, a mention which is the subject of the predicate, and any other non-subject mention associated with the predicate, a Type 1 schema determines the probability of finding the triple in a large corpus with the same mutual relationship. From the sentence "The flower has pollen", Type 1 Predicate Schema <has(m="the flower", a="pollen")> can be extracted. The exact words "has", "flower", and "pollen" must appear in the sentence.
- Type 2: Given two predicates, their subject and object entities, and the connective word which connects their clauses, a Type 2 predicate determines the probability of finding the predicates in a large corpus where their subjects and objects maintain the same mutual relationship. For Type 2 schema, the exact word denoting the subject and object is not important. From the sentence "John is afraid of Jim because he is scared of new people", Type 2 Predicate Schema
  <be_afraid_of(m=*,a=*) | get_scared(m=*,$\widehat{a}$=*), cn="because"> can be extracted. Although exact words "John", "Jim", and "new people" are not needed, their roles to the predicates are.

The authors proposed a method to calculate the scores of each schema that are present in the problem set based on a vector of four different scoring functions. The score S is a vector of $S_{giga}, S_{wiki}, S_{web}$ and S_{pol}, where the "giga", "wiki", and "web" scores are calculated based on the frequency of the schema in Gigaword corpus, Wikipedia corpus, and Google web query, respectively in a similar way as Rahman's approach. The "pol" score is the polarity score that is equivalent of the "Heuristic Polarity" feature of Rahman's approach.

The features of Peng's approach are similar to Rahman's approach, with the exception that Peng's approach does not have lexical features. It achieves better results because it ensures the selection of the candidate with higher score by converting the scores into hard constraints of an ILP and thus overcoming the sparsity problem.

2.2.3 Other Approaches

In recent years, statistical approaches to the Winograd Schema Challenge are based on deep learning using neural networks, sequence ranking and ensemble learning. These approaches have outperformed the previous approaches by a large margin.

Opitz's Approach

Opitz et al. [35] proposed to convert the task of coreference resolution into sequence ranking. For each Winograd Schema problem, two sequences are created by replacing the pronoun with each candidate referent. These two sequences are passed to a deep neural network that assigns a score to each sequence based on the likelihood of the sentence being formed. For example: "The axe cut the tree because it is sharp" creates two sequences.

- The axe cut the tree because the axe is sharp.
- The axe cut the tree because the tree is sharp.

It is clear that the first sequence is more likely to be a valid sentence. After training the neural network on a set of coreference resolution problem, the resulting model is expected to assign higher likelihood for the first sentence than the second one.

Opitz et al. ran two experiments. In the first experiment, the authors split Rahman's problem set into training and testing sets. In the second experiment, all problems of Rahman's set are used to train the model and then the model is used on the Winograd Schema set. The reported accuracy of the model on the Rahman's set and Winograd Schema set is 63% and 54% respectively.

Trinh's Approach

Trinh et al. [50] proposed a method to find the most likely referent using an ensemble of multiple language models. Similar to Opitz et al., Trinh et al. also convert each Winograd Schema problem into two sequences. After that, a language model is trained which calculates

the joint probability of a sequence of text based on their constituent words. The model assigns a score to each sequence which determines the likelihood of the sequence. The authors experimented with two types of scoring.

- Full Scoring: The entire sequence is used for scoring. The score is calculated by joint probability distributions of all the constituent words. For the sentence: "The trophy can not fit into the suitcase because it is too small", S_{full}=P(The trophy can not fit into the suitcase because it is too small).
- Partial Scoring: The entire sequence is split right after the occurrence of the pronoun. The score is calculated by a conditional probability distributions. For the sentence: "The trophy can not fit into the suitcase because it is too small", $S_{partial}$=P(Is too small | The trophy can not fit into the suitcase because it).

The authors proved that partial scoring is more effective than full scoring. Based on this observation, they trained 10 different partial scoring language models on different datasets. The final score is calculated by the ensemble of all 10 language models and is used to determine the correct referent.

Trinh et al. applied their models on two benchmark datasets. The Winograd Schema set $WSCL$ and another smaller set $PDP60$ [8] from the first Winograd Schema challenge. The reported accuracy of the model on the PDP and Winograd Schema set is 70% and 63.7% respectively. However, a critical analysis [49] showed that the main reason for success when using an ensemble of language models was largely due to imperfections in the problem set.

Kocijan's Approach

Kocijan et al. [17] proposed a statistical approach based on language modelling and transfer learning. Kocijan's approach is very similar to Trinh's approach based on the probability model and training procedure. However, while Trinh et al. built the language model from scratch, Kocijan uses Bert's [10] pre-trained model and tuned it on both Rahman's problem set and a custom data set from Wikipedia in order to improve the quality of the model.

To build the custom dataset, the authors implemented a procedure to search a wikipedia corpus for sentences that contain (at least) two occurrences of the same noun. The second occurrence of this noun is replaced with a token that acts as a pronoun. The resulting dataset "MaskedWiki" has similar structure to that of a Winograd Schema problem which is used to fine tune the trained language model to be applied on the original Winograd Schema problem set.

Kocijan's approach outperformed all previous statistical approaches. On the Winograd Schema set, the proposed method achieved 72.5% accuracy. In addition to the Winograd Schema set, the authors also evaluated their model on another problem set $WNLI$ where it achieved 74.7% accuracy.

He's Approach

He et al. [15] proposed a hybrid neral network model. He's approach is an ensemble of Kocijan's approach based on language modelling and Opitz's approach based on sequence ranking. The authors observed the individual strengths and weaknesses of the two approaches and successfully combined them in order to overcome their limitations.

He's approach outperformed both Kocijan's and Opitz's approaches and achieved an accuracy of 75.1% in the Winograd Schema set.

Sakaguchi's Approach

Sakaguchi et al. [43] presented a significantly larger problem sets "WinoGrande" which consists of 43,972 problems similar to the Winograd Schema set. The authors argued that the success of recent statistical approaches are because of having an inherent bias within the Winograd Schema set because it was designed by same group of people. On the other hand, "WinoGrande" was created by crowdsourcing and therefore contains less bias than the official problem set. To prove their argument, the authors trained the BERT [10] model using an approach similar to Kocijan but on a much larger set of problems (43,972 compared to 1886). Their model "RoBERTa-WinoGrande" outperformed all previous approaches and set the current state of the art with an accuracy of 90.1%. However, the accuracy decreased to 79.3% when tested on "WinoGrande". The authors also showed that the accuracy of Kocijan's approach decreases to 59.4% when tested on WinoGrande.

Although, "WinoGrande" is more challenging for statistical approaches, it is still very trivial to even untrained native English speakers. The reported accuracy of Humans on "WinoGrande" is 94.1%, which is significantly higher than the state of the art statistical approach.

2.3 Solutions To The Winograd Schema: Logical Approaches

One of the features of The Winograd Schema is that it is, in the ideal case, "google-proof". Therefore, statistical approaches which can only extract lexical patterns or relatively shallow semantic features like word associativity fail to capture the transferable knowledge or understanding required to solve the Winograd Schema. A number of approaches tries to overcome this limitation by using logical inference on background knowledge. These approaches vary in terms of knowledge structure, extraction method, and inference method.

2.3.1 Schuller's Approach

Schuller et al. [45] stated that while the Winograd Schema itself can be categorized as very hard coreference resolution problem, there is a particular subset of the Winograd Schema problems which are of the highest difficulty in terms of ambiguity. This special

set of problems are different from others because, while most of the Winograd Schema can be solved by semantic analysis only, the special set can not be solved by similar process. Replacing the pronoun with each of the candidate mentions result in semantically valid sentences in its own world of interpretation.

Consider the following Winograd Schema problem: "Pete envies Martin because he is very successful". Here, the word "because" indicates causality and success of an entity is a valid cause of to be envied by other less successful entities. Therefore, semantic analysis can resolve the pronoun "he" successfully to Martin because he is specifically mentioned to be envied by Pete. Changing the word "because" to "although" indicates exception. Therefore, it is also possible for a semantic analyzer to resolve the pronoun to "Pete" by determining a causally unlikely candidate for the pronoun. This type of problem needs only semantic analysis to be resolved.

One of the more difficult problem of the Winograd Schema set is: "Sam's drawing hung just above Tina's and it looked much better with another one below it". Two different interpretations would be:

1. Sam's drawing hung just above Tina's and Sam's drawing looked much better with another one below it.

2. Sam's drawing hung just above Tina's and Tina's drawing looked much better with another one below it

Humans would most likely accept the first option because in the first part of the sentence it is already mentioned that "Sam's drawing is just above Tina's drawing". Therefore, in the second part of the sentence, the other drawing may be Tina's drawing.

However, it is perfectly semantically valid that the other drawing is actually a third drawing, neither Sam's nor Tina's. This interpretation is causally valid and with one additional assumption about the existence of the third drawing below Tina's drawing, the correct answer may change. Therefore, an intelligent agent can also accept the second option with proper logical inference and commonsense knowledge.

Schuller et al. proposed to attack this type of special problem using relevance theory introduced by Schank et al.[44]. Schank introduced a cognitive model for human natural language understanding using knowledge graphs. Schuller et al. proposed to use a simplified model of knowledge graphs to identify the relevance of different interpretations and choosing the more relevant option over less relevant or "far fetc.hed" alternative.

In order to determine the relevance, a fitness function is used which expresses relevance as a function of two cognitive concept: Positive Cognitive Effect(PCE) and Processing Effort(PE).

- Positive Cognitive Effect (PCE): PCE is the change of world representation in the presence of new information. It is a measure of how much understanding of the world is achieved because of new information. It is positively correlated to the relevance.

- Processing Effort (PE): PE is the effort spent on perception, memory and inference. Effort of assuming an existence of a third drawing in the scenario is greater than assuming only Sam's and Tina's drawing. It is negatively correlated to the relevance.

- Relevance (R): Relevance is denoted using the concepts stated above. The relevance function R is calculated as: $R = PCE - PE$

Schuller et al. formally described the knowledge graph structure and how a knowledge graph from an input sentence combines with knowledge graphs of background knowledge. Moreover, they proposed a way to calculate the relevance using a fitness function of how the input graph fits to the background knowledge graphs. The authors worked on four pairs (8 individual) of difficult Winograd Schema problems and experimented with different parameters of the fitness functions.

While Schuller et al. provided a formal procedure to solve even the hardest type of Winograd Schema problems, their method can not be implemented into an automated intelligent system because of the following drawbacks:

1. The approach requires all background knowledge to be in the knowledge graph format. However, no procedure is discussed about how to convert knowledge to knowledge graphs and from where the knowledge can be found. All background graphs are constructed manually.

2. The inference procedure is too complex for any real time system and therefore takes a very long time (almost an hour for a single problem).

Although the method of Schuller et al. is not a general solution, it proves that even the hardest set of the Winograd Schema problems are solvable with formal reasoning. This approach also demonstrates the effectiveness of logical approaches over statistical ones.

2.3.2 Bailey's Approach

Bailey et al. [1] developed correlation calculus and showed its applicability to solve a set of the Winograd Schema. Two sentences or phrases are said to be positively correlated if hearing one makes the other more plausible to the hearer. If F and G are two sentences, positive correlation between F and G is denoted as $F \oplus G$. If prior probabilities of F and G are $P(F)$ and $P(G)$, F and G will be positively correlated if $P(F|G) > P(F)$ and $P(G|F) > P(G)$. On the other hand, F and G are negatively correlated if hearing one make other less plausible or the negation of other more plausible to the hearer. Negative correlation between F and G is denoted as $F \ominus G$. F and G will be negatively correlated if $P(F|G) < P(F)$ and $P(G|F) < P(G)$. The relation of positive and negative correlation is

illustrated below:

$$F \oplus G \equiv \neg F \oplus \neg G$$
$$F \ominus G \equiv F \oplus \neg G \equiv \neg F \oplus G \equiv G \ominus F$$

Bailey et al. claimed that out of the first 100 problems of the Winograd Schema set, 64 exhibit positive correlation and 8 exhibit negative correlation.

The authors proposed an approach which uses background knowledge in the form of first-order logic to infer the correlation between two clauses. Connectives like "because" indicate positive correlation whereas connectives like "although" or "but" indicate negative correlations among the clauses. Replacing the pronoun by each of the candidate mentions in its clause results in two alternative clauses and only one of them can be justified using correlation calculus. The authors worked with three pairs (6 individual) of sentences and showed the effectiveness of their method.

Bailey's approach, like Schuller's approach, does not have any method for knowledge extraction. The background knowledge is created manually. Moreover the solution to the 3 pairs mentioned is not directly applicable to other problems because each of them require different background knowledge. One advantage of Bailey's method is that it uses first-order logic to encode knowledge. There are some existing tools (e.g. OpenCCG [53]) developed by researchers in NLP community that can encode unstructured text to first-order logic and can be used to make the knowledge base. Therefore, Bailey's approach might be more effectively integrated into automated systems compared to Schuller's. However, it is also unknown how large the knowledge base should be and how scalable such a system will be in terms of performance and efficiency.

2.3.3 Sharma's Approach

Sharma et al.[46] developed the first logic-based automated system to solve the Winograd Schema. In the process of solving the problems, the authors developed a semantic parser (K-Parser), a knowledge hunting module, and a reasoning module. Sharma's system can detect the knowledge necessary to solve a Winograd Schema, hunt down the knowledge and use it to reason about the coreference. However, all categories of the Winograd Schema are not solvable with this system. The authors identified two special categories that can be solved. They are described bellow:

- Direct Causal Events: Two causal events, often connected by "because" or "although". The pronoun participates in one event whereas the candidate mentions participate in the other. For example: "The city councilmen refused the demonstrators permit because they feared violence". The events "refuse" and "fear" are connected by connective "because", which shows causal relationship between them. In this example, "fear" causes "refuse".

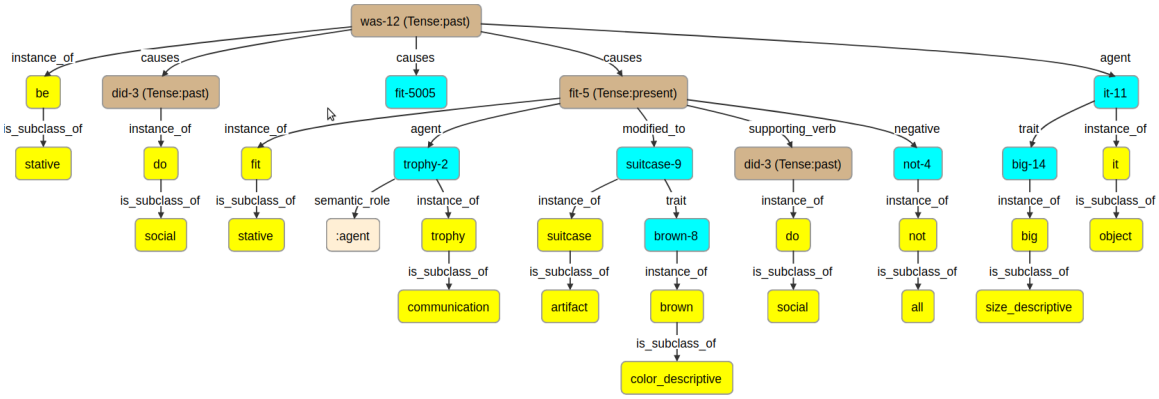


Figure 2.1: A parse tree from "The trophy did not fit into the brown suitcase because it was too big"

- Causal Attributive: One event shared by two candidates. The roles of the candidates are different (e.g, subject and object). The pronoun has a "trait" or "attribute" which is causally related. For example, "The man could not lift his son because he was too weak". The event "lift" is negated by modifier "could not" and connected to the attribute "weak" by causal connective "because". The candidates share the event "lift" and their role is "subject" and "object" respectively. Because of this causal relationship, the trait of the pronoun "weak" can be associated with one of the roles (subject) of the candidates.

Sharma et al. claimed that 71 problems of the Winograd Schema set is covered by these two categories. The authors developed a system to attack the problems in three steps:

1. Semantic Parsing: The authors developed a semantic parser (K-Parser) that parses an unstructured sentence into a semantic tree. All the words of the sentence are parsed as distinct nodes and their relationship is parsed as the edges between the nodes using 14 different semantic roles as edge labels. The nodes are classified as "event" or "entity" and the semantic roles are determined based on PropBank frame sets [36]. Moreover they also provide additional features like word sense disambiguation, coreference resolution, and named entity tagging. A parse tree from the sentence "The trophy did not fit into the brown suitcase because it was too big" is shown in Figure 2.1.

2. Knowledge Hunting: This module creates queries from the parser and use the query to search a large text corpus. The queries are formed by extracting all the events and connectives from the parse tree and joining them by the wildcard character "*". One of the queries from the parse tree shown in Figure 2.1 can be: "did not fit"*"because"*"big". The authors used textual version of world wide web as their corpus. The result is a list of sentences matching the pattern of the query. One of the sentences is used as

the commonsense knowledge sentence and is parsed using the semantic parser in the same way as the Winograd Schema.

3. Reasoning: After the parse tree is generated from both the schema and the commonsense knowledge sentence, the nodes of the trees are matched so that similar chain of events matches. The assumption is that the commonsense knowledge sentences contain no ambiguity, and coreference can be easily determined in the commonsense sentence by the coreference resolution module of the parser. After the coreference is resolved in the commonsense knowledge sentence, Answer Set Programming (ASP) is used to determine the co -reference of the Winograd Schema sentence. It is interesting to note that the accuracy of the coreference module alone on the Winograd Schema is close to a random guess. However, the accuracy is reasonably high on the commonsense sentences due to their easier properties, and therefore the ASP rules can match the nodes of two parse trees and resolve the correct reference in the Winograd Schema.

Sharma et al. used their approach on 71 Winograd Schema problems. The system was able to provide answers for 53 and the remaining 18 was left unanswered. Among the 53 problems, 49 are correct and 4 are incorrect.

Sharma's approach demonstrated reasoning with knowledge and performed better than previous statistical approaches in terms of accuracy (49 correct vs 4 incorrect). However, the approach covered a relatively smaller set.

Sharma et al. criticized Rahman's problem set by pointing out that the corpus contains some redundancy. Sentences like "Lions eat zebras because they are predators" are easier to disambiguate because the pronoun "they" does not agree with both "lions" and "zebras", as a Winograd Schema problem should. Replacing the pronoun by the candidates results in clauses like "Lions are predators" and "Zebras are predators". Because of the fact that, zebras are not predators, the first clause would result in a much higher number of responses than the second. This makes the sentence less ambiguous than a standard Winograd Schema and also violates the "google-proof" property. Therefore, even if the statistical methods works well in Rahman's problem set, they fail to perform equally in the Winograd Schema problem set. Sharma's approach, on the other hand, although covering a very small set of problems, shows much better performance on the Winograd Schema problem set.

In terms of reasoning, Sharma's approach is similar to Schuller's approach because both methods use ASP. However, while Schuller's knowledge graphs are manually generated, Sharma's graphs are automatically parsed from unstructured text. Therefore, this approach can be used to solve a significantly larger set of Winograd Schema (71 compared to 8 for Schuller and 6 for Bailey).

2.3.4 Emami's Approach

Emami et al. [12] proposed a knowledge hunting approach similar to Sharma. However, while Sharma et al. created their own customized knowledge parser, Emami et al.used the Stanford CoreNLP parser to parse both the Winograd Schema sentence and the commonsense sentence. Emami's approach processes a problem instance in four stages.

1. Semantic Representation Schema: The semantic schema is formed using the words of the problem sentence. First, the sentence is parsed using the Stanford dependency parser and the following terms are extracted:

   - E_1, E_2: The candidate referents.
   - $Pred_C$: The context predicate. This is the event that is associated with the referents.
   - $+$: The discourse connective.
   - P: The pronoun
   - $Pred_Q$: The query predicate. This is the event that is associated with the pronoun.

   The semantic schema has the form $<E_1, E_2, P, Pred_C, Pred_Q, +>$. The schema is used for query generation in the next step.

2. Query Generation: the authors assumed that search queries have two components: $Term_C$ and $Term_Q$. These are the strings which represent the events associated with the first and second clause of the sentence. Based on this assumption, two sets of queries are constructed, C and Q. Any event, adjective, and root verb can go to either C or Q based on their association with the referents and pronoun respectively.

   Once the queries are constructed, they are augmented with WordNet synonyms to increase the number of search results.

3. Parsing Search Results: Each sentence returned by the queries is called an evidence sentence. Evidence sentence can have one of the four forms:

   (1) $E_1\ Pred_C\ E_2 + E_3\ Pred_Q$

   (2) $E_1\ Pred_C\ E_2 + Pred_Q\ E_3$

   (3) $E_1\ Pred_C + E_3\ Pred_Q$

   (4) $E_1\ Pred_C + Pred_Q\ E_3$

   The evidence sentence may contain coreference. The Stanford CoreNLP framework is used to resolve the coreference in the evidence sentences. Usually, coreference resolution in sentences from web queries are significantly easier than the Winograd Schema.

Therefore, the Stanford CoreNLP can reliably disambiguate such sentence. The following sentence can be considered: "She tried to call him but she was not successful". Here, the sentence is of form (i) and can be parsed as the following: E_1="She", $Pred_C$ ="tried to call", E_2="him", +="but", E_3="she", $Pred_Q$="was not successful".

4. Antecedent Selection: After the evidence sentence is extracted, the parser determines if the verb is in active or passive voice. The correct referent or antecedent is determined using evidence label and evidence strength. Evidence can have one of the two labels: "evidence-agent (EA)" or "evidence-patient (EP)". The labels of the evidence $L(e)$ are determined using the following rules:

$$L(e) = \begin{cases} EA & \text{if } E_3 \text{ refers to } E_1, \text{ active} \\ EA & \text{if } E_3 \text{ refers to } E_2, \text{ passive} \\ EP & \text{if } E_3 \text{ refers to } E_2, \text{ active} \\ EP & \text{if } E_3 \text{ refers to } E_1, \text{ passive} \\ EP & \text{if } E_3 \text{ refers to } E_1, \text{ causative} \end{cases}$$

The strength of the evidence is calculated based on length score and order score. The length score is calculated as:

$$LenScore(e) = \begin{cases} 2 & \text{if } len(Term_Q) > 1 \\ 2 & \text{if } len(Term_C) > 1 \\ 1 & \text{otherwise} \end{cases}$$

$$OrderScore(e) = \begin{cases} 2 & \text{if } Term_C \text{ appears before } Term_Q \\ 1 & \text{if } Term_Q \text{ appears before } Term_C \end{cases}$$

Finally, the strength of the evidence $Str(e)$ is calculated as: $Str(e) = LenScore(e) + OrderScore(e)$

Emami's approach is more robust than Sharma's approach and covers 273 problems of the Winograd Schema set. However, it has a lower accuracy than Sharma's approach. Emami's approach can solve 119 problems correctly and provides no answer for 74 problems. The authors provided additional mechanism for random guessing which can correctly solve 37 more problems. The reported accuracy for Emami's approach is 57.1%

2.4 Solutions To The Winograd Schema: Combined Approaches

The statistical methods of coreference resolution are highly scalable and efficient. However, they only extract lexical properties and shallow semantic compatibility. On the other hand, logical approaches, in spite of showing capability of performing logical inferences, fail to generalize and show relatively poor scalability and efficiency. Because of these trade-offs, some approaches try to combine both statistical and logical methods.

2.4.1 Liu's Approach

Liu et al. [21] won the Winograd Schema Challenge of 2016. Their method used vector embeddings for words and phrases to calculate the likelihood of a candidate mention to be the referent of a pronoun. They applied their systems on the Pronoun Disambiguation Problem (PDP) and the Winograd Schema. Their approach can be used for an arbitrary number of pronouns and candidate mentions. For any Winograd Schema problem, the context plays an important role. One of the key assumption is that if a pronoun refers to a mention, replacing the mention in the place of the pronoun would result in no semantic change of the sentence.

The authors proposed to inject commonsense knowledge as constraints while creating the word vector embeddings. This approach is similar to Peng's approach. However, the nature of commonsense knowledge is different. While Peng's method uses frequency counts of triplets from google and wikipedia, Liu's method uses semantic knowledge bases like ConceptNet [20], WordNet [29], and CauseCom [21] to determine the Knowledge Enhanced Embeddings (KEE) for words. These knowledge bases contain more reliable and refined knowledge compared to frequency counts from wikipedia and google.

The constraints are generated as follows:

- ConceptNet: ConceptNet stores a large set of words and their relationships. The words are considered as concepts and the relationships indicates how these concepts link together. For any word u, if v is a word having a relation with u, and w is a word having no relation with u, $dist(u, v) > dist(u, w)$.

- WordNet: WordNet stores commonly used words along with their synonyms and antonyms. For any word u, if v is a synonym of u, and w is an antonym of u, $dist(u, v) > dist(u, w)$. For example, the word "good" and "nice" must be closer than the word "good" and "evil".

- CauseCom: CauseCom is a knowledge base which stores cause and effect pairs as directed graphs in which the edge between two nodes indicates the presence of a cause and effect relationship. The weights associated with edges indicate the confidence of the relationship. For any cause and effect pair, the distance between them must be proportional to their weight.

Word	Left	Right
The trophy	-	did not fit into
The suitcase	did not fit into	because it is too
It	into the suitcase because	is too big

Table 2.2: The contexts in Liu et al's approach for the sentence: "The trophy did not fit into the suitcase because it is too big"

The KEE for individual words are used to encode left and right contexts for each of the candidates and pronouns. For the sentence, "The trophy did not fit into the suitcase because it is too big", the contexts would be similar to Table 2.2.

The contexts are of varying length. Therefore, they are further encoded to fixed length representations and then projected to the vector space using the KEE of their words. Because of the commonsense knowledge used to create the KEE, the representations of the left and right contexts will contain the semantic knowledge and therefore similar contexts will be embedded nearby in the vector space. For any Winograd Schema problem, distance is calculated between the context of a pronoun and the contexts of each of the candidates. The candidate with the shortest distance is selected.

The authors applied this approach to both the PDP set and the Winograd Schema set. In the PDP set, it was able achieve an accuracy of 61.7%. However, in the Winograd Schema set, the accuracy was 52.8%, slightly better than random guess.

Some of the major contributions of the authors are:

- The method is very scalable and completely automated, therefore well-suited to be used for intelligent agents.
- The sparcity problem of knowledge is one of the main challenges of knowledge extraction. Previous approaches like Sharma's approach depend on finding exact matches for all pairwise verbs or events or traits. Finding exact words takes a long time and requires a large search space. Because word embeddings are continuous vectors, it is not required to find exact matches in order to find useful knowledge. For example, cause and effect pairs like "rob" and "be arrested" can serve as useful knowledge for other similar pairs like "crime" and "be punished" as well.

Even though Liu et al. was successful to incorporate reliable and refined knowledge from established knowledge bases and used them more effectively and efficiently, the method still fails to achieve significant improvement for the general set of the Winograd Schema.

2.5 Human Performance

In the previous sections, performances of Artificially Intelligent (AI) programs on the Winograd Schema problems are discussed. This chapter will contain the human performances on

the same problem set. Human performances can serve as an upper benchmark for the AI programs. Bender et al. [2] ran several experiments on the performance of humans in solving the Winograd Schema and established a human baseline. The experiments were run on 430 humans of different ages using a set of 160 Winograd Schema problems. Among the 160 problems, 17 were classified as easy and the remaining 143 were classified as hard. The participants were selected based on their performances in a training round where they were presented with a set of 7 easy and 1 hard problems. The actual testing round includes 36 hard and 4 easy problems, randomly selected from the original set.

Among the 430 participants, 23 did not finish the test. From the remaining participants, the accuracy of the testing round was 92.1% with $\sigma = 0.07$. Within only easy problems, the accuracy increases to 98.6% with $\sigma = 0.08$.

On average, each of the problems was answered 50.9 times. 75% of the problems was answered correctly by at least 90% participants and 17% was answered correctly by 100% participants. 26 problems had accuracy between 80% and 90%, and only one had accuracy less than 50%.

The experiment provided the participants an opportunity to comment on the test and explain possible causes of mistakes. The responses are recorded for judging the difficulty of the problems. Some of the main reasons for mistakes are:

- Ignoring the context and depending on experience:

  "My meeting started at 4 and I needed to catch the train at 4:30. So, there was not much time. Luckily, it was delayed. So, it worked out."

  This problem contained several sentences and sub sentences. The correct answer is "the train". However, some of the participants focused only on "It was delayed" and answered "the meeting" by reasoning that "Meeting was delayed" is far more likely than "Train was delayed".

- Dependency on personal preference:

  "Everyone really loved the oatmeal cookies; only a few liked the chocolate chip cookies. Next time, we should make more of them"

  For this problem, the correct answer is: "oatmeal cookies". However, some participants incorrectly answered "chocolate chip cookies" because they can not think of a scenario in which anyone can prefer oatmeal cookies to chocolate chip cookies. According to one of the participants, the only type of cookies worse than the oatmeal cookies are raisin cookies!

- Subjective opinion on controversial topic:

  "Pam's parents came home and found her having sex with her boyfriend, Paul. They were embarrassed."

The correct answer would be “Pam and Paul”. However, some participants, specially older ones, answered “Pam’s parents” because it is very embarrassing for parents to see their children having sex.

- Semantic similarity between the special and alternative word: A particular problem was answered correctly by less than 50% of human participants. It was:

  ”I could not put the pot on the shelf because it was so tall”.

  The correct answer would be “The pot”. The reason of the ambiguity is that human participants equated the word “tall” with the word “high”. The problem was designed with the word “tall” and “high” as the special and alternative words such that switching the word changes the correct answer. When presenting the problem along with its alternative, the ambiguity became easy to resolve. However, in Bender’s experiment, the participants looked at one problem at a time and therefore it became harder to resolve. The twin of the sentence: “I could not put the pot on the shelf because it was so high” was much easier to solve and was correctly answered by more than 95% of the participants.

2.6 Machine Reasoning vs Human Reasoning

Human performances on the Winograd Schema are still better than even the best state of the art AI programs on unbiased Winograd Schema problems. Table 2.3 illustrates a brief comparison of previous approaches and human performance on the Winograd Schema set. The difference of performance proves the cognitive superiority of human-like reasoning over AI programs. Recently, the difference between human reasoning and machine reasoning has gained some interest among researchers. Richard et al. [40] claimed that state of the art automated reasoning systems often fail to consider the role of pragmatics in solving the Winograd Schema Challenge. Both the statistical and logical approaches focus on mining of commonsense rules from existing knowledge bases or test corpus instead of learning transferable knowledge or understanding. Theses rules are based on the probability of two or more events occurring together in a sentence along with their relationships with subject and object entities. A few cases that still have a lot of room for improvement are:

- Reasoning about an unknown object based on properties:

  “The large ball crashed through the table because it was made of steel”

  The above coreference problem can be solved by existing methods by reasoning that something made of steel is more likely to crush other things than be crushed. However, if the word “steel” is replaced by an unknown word “XYZZY” along with some additional information about “XYZZY” (e.g., 98% air, lightweight, etc.), the problem becomes impossible to solve by current methods. The human-like reasoning would be

"steel is hard and anything that is hard is likely to crush" instead of "steel is likely to crush ".

- Reasoning about an unknown event-entity pairs based on roles:

  "Can a deer run steeplechase?"

  The above question is not answerable by existing methods because there is hardly any evidence of any deer running steeplechase. However, the human-like reasoning would be: "Deer can jump and anything that can jump can run steeplechase".

Approach	Technique	Coverage	Accuracy
Sharma	Customized Parser + Query + ASP	71	49 Correct, 4 Wrong, 18 None
Emami	Stanford Parser + Query + Rule Based Scoring	273	119 Correct, 80 Wrong, 74 None
Rahman	Statistical Features + Sequence Ranking	Different Set	-
Peng	Statistical Constraints + ILP	Different Set	-
Liu	Word Embedding + Commonsense	273	52.80
Opitz	(Word + Edge) Embedding + BiLSTM	282	54.00
Trinh	Language Models + LSTM + Ensemble	273	63.70
Kocijan	Language Models + Transfer Learning on Rahman's Set (1886 samples)	273	72.50
He	Kocijan + Opitz	273	75.10
Sakaguchi	Language Models + Transfer Learning on WinoGrande (43972 samples)	273	90.10
Human	-	160	92.10

Table 2.3: Performance Comparison: Previous Works vs Human

2.7 Works On Semantic Classification and Polarity

Our approach to the Winograd Schema Problems is based on semantic classification of verbs and adjectives. To the bast of our knowledge, this is the first step towards verb classification based on multi dimensional polarity. However, there are a few works which deal with multi dimensional adjective classification.

Feature	Pos	Neg	Neu	Tot
Emotion	52	73	3	128
Moral/Ethic	45	155	2	202
Character	355	584	220	1159
Weather	7	25	6	38
Color	0	9	42	51
Quantity	16	0	9	25
Appearance	41	83	46	170
Material	22	11	54	87
Shape	0	0	30	30
Touch	3	13	6	22
Taste	40	41	5	86
Dimension	11	2	60	73
Chronologic	3	0	30	33
Geographic	0	10	19	29
Others	29	17	87	133
Total	624	1023	619	2266

Table 2.4: The features used by Tuveri et al.

2.7.1 Tuveri's Classification

Tuveri et al. [51] proposed the multi dimensional polarities of common adjectives. The authors argued that polarity of an adjective is in fact a combination of different features and listed 14 of such features. Each feature has its own positivity or negativity and may be independent or dependent on other features. Based on this observation, a set of 2266 adjectives was selected and evaluated manually by two human evaluators. The results obtained by the evaluators were compared to measure the disagreement. If a convergence of opinions was not possible, the result was discarded. The results are summarized in Table 2.4.

2.7.2 Personality Classification

Majumder et al. [51] proposed a deep learning based method for classification of personalities. The adopted model is based on the big five personality model [14] where personalities are assessed in five categories: "Openness", "Conscientiousness", "Extroversion", "Neuroticism", and "Agreeableness". In order to calculate scores in each of these five categories,

the authors created a set of features for personality detection and classified 14,183 English words based on the set. The features are: “anger”, “anticipation”, “disgust”, “fear”, “joy”, “negative”, “positive”, “sadness”, “surprise”, “trust”, and “charged”.

2.7.3 Works on Aspect Based Polarity Detection

Aspect Based Sentiment Analysis (ABSA) is the task of classifying sentiment polarity with respect to a set of aspects. In ABSA, polarity of a text is represented as a vector of polarity features. Motivated by the success of deep learning in representation learning, Dong et al. [11], Lakkaraju et al. [18], Nguyen et al. [33], and Wang et al. [52] proposed deep learning based methods to generate sentence embeddings and using the embedding to classify a sentence or text sequence into either “positive”, “negative”, or “neutral”. Wang et al. enhanced the embedding with an attention mechanism which applies discriminative weights on different part of the text sequence.

A variation of sentiment analysis is called Targeted Sentiment Analysis (TSA). While ABSA aims to determine the polarity of a sentence or text sequence, TSA determines the polarity of specific entities of text sequences. Some of the interesting works on TSA have been presented by Tang et al. [47] and Chen et al. [6]. These methods are based on Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) and use attention mechanism to refine important words from a text sequence.

Saeidi et al. [42] combined ABSA and TSA to create a system for Targeted Aspect Based Sentiment Analysis (TABSA). The authors proposed a feature-based logistic regression model and a LSTM-based model.

The SenticNet project [3] provides several knowledge bases for sentiment classification and polarity detection. Ma et al. [23] proposed an efficient method for TABSA using Bidirectional LSTM (BiLSTM) and encorporating the commonsense knowledge from the SenticNet knowledge base.

Among all the approaches described above, our objective is very similar to Ma et al. To solve the Winograd Schema, we need to target the pronoun and candidate referents. Therefore, we can make them the targeted words. After extracting the polarities of both the pronoun and the candidates, we can compare them to align the pronoun to the correct referent. This is the main idea behind our approach. However, Ma et el. used a supervised approach which relies on the knowledge of SenticNet and also the training examples of SentiHood [42] and Semeval [38]. The examples of both the datasets are mostly reviews of commercial products and therefore not enough for the Winograd Schema Challenge. For example: some of the aspects are “price”, “taste”, etc. Compared to these aspects, we found the classification of Tuvery et al. [51] more generic, and we used that as a base for our polarity vector. Because of this difference, the training data from SentiHood and Semeval are not very useful in solving the Winograd Schema. Therefore, we need to create our own

commonsense knowledge base based on Tuveri's polarity features. Moreover, as we can not use the existing training data, we need a method which can perform without any training.

Chapter 3

Methodology: Construction of the Knowledge Base

From chapter 2, it is evident that the Winograd Schema Challenge requires semantic analysis and logical reasoning. In order to perform logical reasoning, we need a set of ground truths and a logical inference model. In our approach, we create a knowledge base to serve as the ground truth. Because of the large diversity of the domain of the Winograd Schema, we need many different forms of reasoning (e,g., causal, temporal, epistemic). In this thesis, we limit our study within only causal reasoning. The problems which require causal reasoning usually contain connectives like "because" or "although". However, existence of such connectives is not guaranteed. An example is: "The fish ate the worm; it was hungry". Even though no connective is present, causal reasoning is required because the adjective "hungry" is the cause of the event "eat".

In order to perform the logical reasoning, we need to store semantic information for both "hungry" and "eat". Moreover, as discussed in chapter 1, both the pronoun and the candidate referents must be associated with at least one event and the event must contain at least one root verb. Therefore, we only need to store lexical semantic information about verbs and adjectives.

It is important to identify the type of semantic knowledge to be stored for the verbs and adjectives. Currently, one of the most effective methods to store information about words is to use their vector embeddings [28]. However, the vector embeddings constructed by Bag-of-Words [27] and Skip-gram [27] are constructed based on the contexts in which they appear and therefore words that are semantically different may have similar vector representation as long as they appear in similar context. For example, the words "good" and "bad" appears close to each other in Word2Vec because of their applicability in similar context, regardless of their opposite semantic meaning. Moreover, the information encoded in vector embeddings is mostly syntactical and contains very little semantic information.

As an alternative, we store the emotional or sentiment polarity as the semantic information. In this chapter, we discuss different types of sentiment polarity in common adjectives

and verbs and how the knowledge base is constructed. Sentiment polarity has a long history of being used in NLP problems. Researchers like Nasukawa et al. [32] and Wilson et al. [54] described polarity as positive or negative opinions. Some other researchers like Pointiki et al. [38], Wang et al. [52] and Thet et al. [48] broke down the concept into multiple aspects. For solving the Winograd Schema, the multi-aspect model is more effective because is allows a single word to have different polarity values for different aspects and each aspect can have different features [48]. We consider polarity of words as a multi-dimensional vector where each dimension represents a feature with positive, negative or neutral value. Different researchers defined sentiment polarity in different ways. To avoid confusion, we define polarity as follows:

Definition 1. *Polarity: The polarity of a verb or adjective is a vector of features where each feature represents a positive, negative, or neutral emotion which can be felt by an unbiased human about the word itself (adjective) or the subject of the word (verb). The features can be independent or dependent.*

Our definition of polarity falls into the category of target aspect based sentiment analysis (TABSA). Also, in this thesis, we use "aspect" and "feature" interchangeably. Most of the recent aspect based polarity analysis methods are domain specific. For example, Thet's [48] method is targeted to the domain of movie reviews and therefore, the selected aspects or features were "cast", "music", "character", etc. For the Winograd Schema, we want a generic set of features that are domain independent.

Representing polarity as a multi-dimensional vector allows us to generalize the works of [39] and [37] where polarity is expressed as one dimensional with the value "positive", "negative", or "neutral". In this chapter, we discuss the construction of the polarity vector as well as the knowledge base.

3.1 Polarity Feature Vector Construction

In chapter 2, we discussed Tuveri et al.'s classification of polarity and described their feature vector in Table 2.4. One of the major challenges of applying the classification for solving the Winograd Schema is the interdependency of the features. For example, "moral/ethic" and "character" are closely related and the polarity of one of them influences the polarity of the other. For the feature vector, we want the features to be as independent as possible. Therefore, we convert the features of Tuveri into a set of independent features.

We analyze several types of possible relations among the features. To detect similarity between two polarity feature, we considered using Jaccard's Similarity Coefficient [16] similar to the work of [34]. In our approach, let us consider two features p and q. If the number of words for which p and q have non-neutral value is n_p and n_q respectively, and the number

F1	F2	F3	F4
1	1	1	-
0	0	0	0
1	1	1	1
0	0	-	0
1	0	-	1
1	-	1	1
1	-	-	1
-	1	-	0
0	-	0	0
0	-	-	0

(a) Example Data and Features

	F1	F2	F3	F4
F1	1	4/9	5/9	8/9
F2	4/6	1	3/6	3/6
F3	1	3/5	1	4/5
F4	8/9	3/9	4/9	1

(b) Agreement Matrix: Example Data and Features

	eth	emo	ch	wt	ap	mt	tc	ts	dm
eth	1	0.23	0.33	0.07	0.1	0.03	0.01	0.01	0
emo	0.28	1	0.92	0.06	0.13	0.02	0	0.01	0
ch	0.31	0.71	1	0.03	0.13	0.02	0	0.01	0
wt	0.4	0.28	0.16	1	0.6	0.16	0.04	0.08	0
ap	0.26	0.28	0.35	0.26	1	0.19	0.04	0.04	0.05
mt	0.23	0.09	0.14	0.18	0.5	1	0	0.09	0
tc	0.2	0	0	0.2	0.4	0	1	0	0
ts	0.33	0.33	0.33	0.67	0.67	0.67	0	1	0
dm	0	0	0	0	0.75	0	0	0	1

(c) Agreement Matrix: Wilson's Lexicons and Tuveri's Features

Table 3.1: Agreement, dependency, and independency of feature vectors

of words where they have the same value is n_{pq}, Jaccard's Similarity Coefficient is:

$$J(p,q) = \frac{n_{pq}}{(n_p - n_{pq}) + (n_q - n_{pq}) + n_{pq}} \tag{3.1}$$

It is important to note that Jaccard's Similarity Coefficient is symmetric and therefore $J(p,q) = J(q,p)$. For analyzing the features, we want to identify not only the related features, but also which features are influencing others. Therefore, we modify Jaccard's Similarity Coefficient and define a new similarity metric, which we define as "agreement". Based on "agreement", we measure "equivalence", and "independence" which are defined as follows:

Definition 2. *Agreement: The agreement of p to q and q to p is denoted by $A(p,q)$ and $A(q,p)$ respectively and defined as:*

$$A(p,q) = \frac{n_{pq}}{n_p} \tag{3.2}$$

$$A(q,p) = \frac{n_{pq}}{n_q} \tag{3.3}$$

Table 3.1(a) shows an example with four polarity features: F1,F2,F3, and F4 and 3.1(b) shows the corresponding agreement values. Each row indicates a different word and "1" and "0" indicate positive and negative polarity values respectively. Among the 10 rows (words), Feature F1 has 9 non-neutral values and F2 has 6 non-neutral values. Non-neutral values for F1 and F2 are same in four rows (first four). Therefore, the agreement of F1 to F2 and F2 to F1 is 4/9 and 4/6 respectively.

Definition 3. *Equivalence: Two features p and q are considered equivalent if*

$$min(A(p,q), A(q,p)) \geq \alpha \tag{3.4}$$

Here, α is a threshold determined manually. In Table 3.1(a) and 3.1(b), for $\alpha = 8/9$, F1 and F4 are equivalent features.

Definition 4. *Independence: Two features p and q are considered independent if*

$$max(A(p,q), A(q,p)) \leq \beta \tag{3.5}$$

Here, β is a threshold determined manually. In Table 3.1(a) and 3.1(b), for $\beta = 3/5$, F2 and F3 are independent features.

Definition 5. *Active and Passive Agreement: Feature p actively agrees with feature q if*

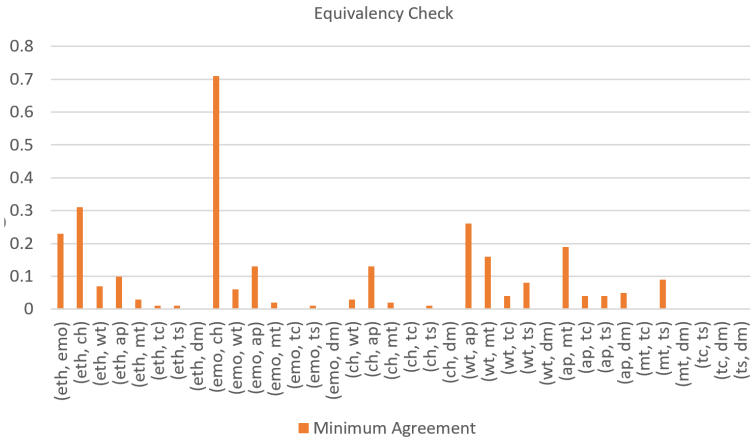
$$A(p,q) \geq \alpha \geq A(q,p) \tag{3.6}$$

In Table 3.1(a) and 3.1(b), for $\alpha = 8/9$, $A(F3, F1) \geq \alpha \geq A(F1, F3)$. Therefore, $F3$ actively agrees with $F1$ and $F1$ passively agrees with $F3$. Active and passive agreement indicates cause and effect relations among features. If we look at Table 3.1(a), we see that whenever $F3$ has any non-neutral value, $F1$ has the same value, but not the other way round. Therefore, it is highly likely that $F3$ is influencing the value of $F1$ or $F3$ is a cause of $F1$. In other words, $F1$ is a feature with many causes and $F3$ is one of them. Example of such feature pair can be "character" and "kindness", where positive or negative value in "kindness" is a cause of similar value in "character". We need to eliminate this type of relationship because it hinders the independence of features.

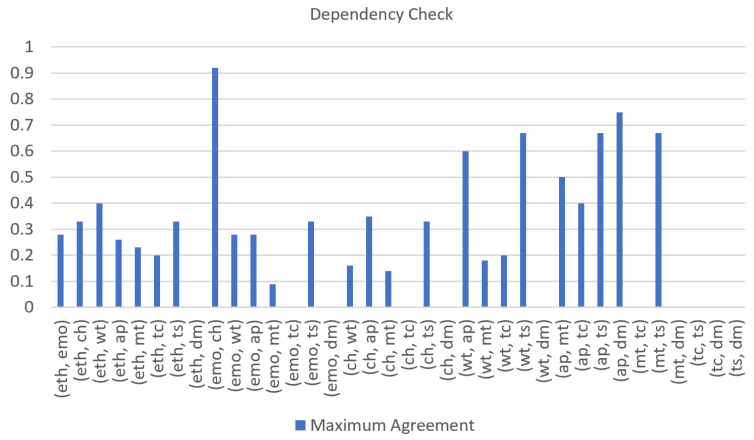
Our computation of the above metrics on Tuveri's features are shown in Table 3.1(c). We convert the features of Tuvery into an independent set by removing redundant features, merging dependent features, and shrinking the feature vector if necessary.

3.1.1 Removing Redundant Features

There are some features in Tuvery's feature set which can never have certain values. For example, "color" can never have positive or negative value and is always neutral. We iden-



(a) Determining α (Sample Size=36, Mean=0.08, Standard Deviation=0.13)



(b) Determining β (Sample Size=36, Mean=0.27, Standard Deviation=0.25)

Figure 3.1: Manual determination of the thresholds α and β

tify and remove these features because they can not provide any useful information to solve the Winograd Schema. These are: “color”, “quantity”, “shape”, “chronologic”, and “geographic”. The remaining features are: “emotion”, “moral/ethic”, “character”, “weather”, “appearance”, “material”, “touch”, “taste”, and “dimension”.

3.1.2 Merging Equivalent Features

We remove inter-dependency from the remaining features by merging the equivalent features. In order to merge the equivalent features, we need to determine the values of the thresholds α and β. We use the subjectivity lexicon of Wilson et al. to determine these thresholds. The lexicon contains 8222 words (2170 nouns, 3250 adjectives, 1350 verbs, 330 adverbs, and 1147 other). We sample a small set of 374 adjectives and determine the values of their polarity features manually. Next, we compare our polarity values with Wilson’s and

analyze the polarity features. Moreover, for each pair of features, we compute the agreement matrix shown in 3.1(c). After computing all the agreement values for each pair of features, we determine the active and passive agreements by calculating the maximum and minimum agreement values between them. In order to determine equivalence, we use the minimum agreement values as shown in Figure 3.1(a). We observe that "emotion" and "character" have the highest minimum agreement (0.71), which is significantly higher than sample mean (0.08). Therefore, we set the value of $\alpha = 0.7$ and consider these two features equivalent. After detecting the equivalent features, we merge them together. In this step, "emotion" and "character" were combined together.

In order to determine which features are dependent on other, we use the maximum agreement values as shown in Figure 3.1(b). We observe that most of the values are lower than 0.5, and therefore set the value of our threshold $\beta = 0.5$. For this value, we have 7 pairs of dependent features. However, except "emotion" and "character", other 6 pairs have minimum agreement values lower than 0.3, and therefore are only in passive agreement (one is influencing the other). This type of relationship is useful to create a hierarchical structure. However, in this step, we do not merge these feature pairs.

3.1.3 Shrinking Feature Vector

After performing the first two steps, we have a vector of 8 features. If we consider all 8 features and 3 values for each feature (positive, negative and neutral), we can have 3^8 possible states. Our manually marked set of 374 adjectives is not sufficient to represent all of the states. Moreover, some features (e.g., weather) have neutral values for most of the words. Therefore, in this study, we experiment with smaller sets of features and compare their performances. Experimenting with longer feature vectors remains one of the possible future works.

In order to shrink the size of the feature vector, we identify the features which are more informative than others. For example, "moral/ethic" is more important than "dimension" and therefore, any word having negative value in "ethic" and positive value on "dimension" (e.g., monstrous) will almost always have an overall negative polarity. In order to form a polarity vector of size k, we need to sort the feature vector according to their priority and select only the first $k - 1$. The remaining features, although independent, can be merged together as a single feature and the value can be determined by majority voting.

We determine the priority of the features using 2 methods: i) agreement matrix and ii) neural network weights.

- Agreement Matrix: We use the polarity assigned by Wilson et al. as the ground truth and perform the following steps:

    1. Start with feature list F, word list W and priority list $P = \emptyset$.

2. Find feature $f \in F$ which has maximum agreement with the ground truth (Wilson's Polarity).
3. Add f to P and remove f from F
4. Remove from W all the words which have non-neutral value for F so that new W represents only the words whose polarities do not depend on f
5. Follow the steps 2 to 4 until either W or F becomes empty.

- Neural Network Weights: We use a single layer perceptron using Wilson's polarity as the target class. The target can be either "positive", "negative", or "neutral". The input is constructed by the feature vectors of a set of 374 adjectives which we annotated manually. The perceptron is trained to produce output values similar to the target for each adjective. Our objective is to observe how much weight is applied on each feature in order to make the output similar to the target. We use the following parameter for training the perceptron: $[epochs = 1500, \alpha = 0.01, activation = sigmoid]$. After training, we sort the features according to their weights.

The priorities calculated by both procedures are shown in Table 3.2. The agreement matrix results in

ethics > *character* > *weather* > *touch* > *dimension* > *appearence* > *material* > *taste*

whereas the neural network weights results in

ethics > *character* > *weather* > *appearence* > *touch* > *material* > *taste* > *dimension*

eth	ch	wt	tc	dm	ap	mt	ts
0.77	0.76	0.60	0.40	0.36	0.25	0.25	-

(a) Agreement Matrix

eth	ch	wt	ap	tc	mt	ts	dm
5.01	4.51	3.21	2.19	1.34	0.35	-0.88	-1.15

(b) Neural Network

Table 3.2: Priorities of the features

In this thesis, we worked with only $k = 1$ and $k = 2$. When $k = 1$, we call the polarity feature as "level 1" polarity. Level 1 polarity is a single value which carries the same meaning as Wilson et al. When $k = 2$, we call the feature vector "level 2" polarity. Level 2 polarity has two features. The first one is "ethics" which is the first member of the priority list. The second one is a combined feature obtained by merging all features of the priority list except "ethics". In this thesis, we call the combined feature "ability".

3.2 Adjective Classification

For both level 1 and level 2 polarity, we classified a larger set of 436 adjectives manually. We used this set as our initial knowledge base. For level 1, the polarity values of Wilson are used directly.

However, this polarity classification is prone to errors. For example, some adjectives like "hard" have a negative value in Wilson's classification. In practice, "hard" may have either a positive or negative effect on humans mind, depending on context. The following examples illustrate:

- Alex cannot be persuaded to lie, he is too hard.
- Alex cannot be persuaded to help the homeless, he is too hard.

In the first example, the hardness of Alex imposes a positive impression and in the second sentence the same adjective imposes a negative impression. It is therefore not sufficient to judge Alex by level 1 polarity. For level 2, we have two features: "ability" and "ethics". We can assign a negative value in "ethics" and a positive value in "ability" for the adjective "hard" and thus explain both of the cases effectively.

The larger the size of the polarity vector, the better would be the classification of adjectives. However, it also involves higher number of possible classes which increases exponentially with the number of features and may make the classification infeasible. If we have k features and each have three possible values (positive, negative, or neutral), any adjective can classified into one of the 3^k classes. Moreover, features located near the end of the priority list (e.g., touch, taste) are very sparse in nature and therefore are applicable to a very limited set of adjectives. This introduces a trade-off between the accuracy and feasibility.

For the level 2, we have "ability" and "ethics". Figure 3.2 shows some examples of level 2 classification of adjectives.

3.3 Verb Classification

A verb in a sentence is used to indicate an action, and typically has a subject and an object. A subject, also called an agent, is an entity who performs the action and an object is another entity upon whom the action is performed. In order to classify verbs, we consider their applicability and the type of subjects and objects associated with them. The polarities of the subjects and objects determine the polarity of the verb. If the verb is compatible with entities having conflicting polarity values, we set the polarity as "neutral" or "don't care".

If we consider only the level 1 polarity, we have only one feature for the polarity. Based on the polarity values of subject and object, the verb can be one of the following types:

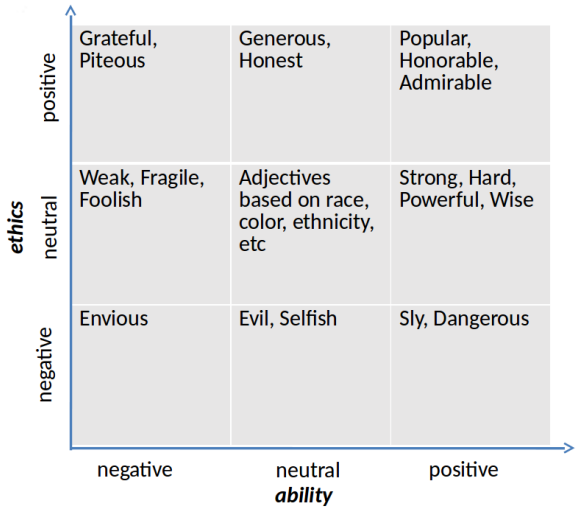
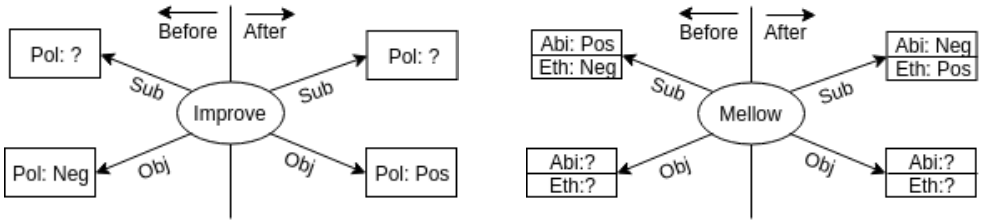


Figure 3.2: Classification of adjectives based on "ability" and "ethics"

- Positive subject, positive object: "Alex *rewards* Bob for his honesty". In this example, the entity who rewards and the entity who is rewarded have positive polarity. Therefore, the verb "reward" belongs to Positive-Positive class.
- Positive subject, negative object: "Alex *punishes* Bob for his dishonesty". To be punished invokes negative sentiment and therefore the object have negative polarity. The verb "punish" belongs to Positive-Negative class.
- Negative subject, positive object: "Alex *envies* Bob for his success". In this example, the polarity of the subject is inferior than the polarity of the object. Therefore, the verb "envy" belongs to Negative-Positive class.
- Negative subject, negative object: "Alex $fools$ Bob and steals his money". This is an example where both subject and object entities have negative impression on human mind. The verb "fool" belongs to Negative-Negative class.
- Other: where polarities of subject or object, or both are neutral or unknown: "Alex *drives* a car". We can not classify "drive" to any single category. This type requires additional analysis.

For the particular case where the polarity is neutral or unknown, we simply assume that verb to be compatible with all possible polarity values of the subject or object. In the sentence, "Alex *drives* a car", it does not matter if Alex is a thief, a charity worker or a revolutionary scientist. The polarity of the subject is not important for the verb "drive". In our classification, we add "drive" to all four classes mentioned above. This categorization makes the classes overlap. However, it prevents our approach from creating too many unnecessary, near empty classes.



(a) Level 1 polarity of "Improve" at "Alex's skill was improved"

(b) Level 2 polarity of "Mellow" at "Alex mellowed in his later days"

Figure 3.3: Classification of verbs using features from Level 1 and Level 2 polarity

Classification based on only subjects and objects are not enough to understand the causal property of the Winograd Schema problems. There are many examples of events which denote actions that change the polarities of subjects and objects. Based on the values of the polarity vectors before and after the action, we observe two types of verbs:

- Static Polarity: "Alex *knows* about history because he is knowledgeable". The verb *know* does not change polarity of its subject or object.
- Dynamic Polarity: "Alex was naive before Bob *taught* him and became expert after". The verb *teach* changes polarity of its object from negative to positive. However, the polarity of its subject is always positive. We call the polarity before the event as the "pre polarity" (Pol_{pre}) and the polarity after the event as the "post polarity" (Pol_{post}).

If we consider the dynamic polarity, the values of the polarity vectors can change to any of the four types (positive-positive, positive-negative, negative-positive, negative-negative) after the event. Therefore, when we perform the classification of verbs based on both subject-object polarity and static-dynamic property based on the level 1 polarity, we end up with 16 possible classes. Any verb can be in one or more of these 16 classes.

Classification of verbs using level 2 polarity features (i.e, "ability" and "ethics") based on both subject-object polarity and static-dynamic polarity results in 256 possible classes. Although this type of classification involves a lot of sparse classes, it is more effective than using only level 1 features in many cases. For example, in a sentence: "Alex's skill was improved", dynamic events like "improve" can be easily classified using only level 1 polarity. However, a less obvious example is "Alex considerably mellowed in his older age" where dynamic event like "mellow" requires analysis of the polarity from perspective of both "ability" and "ethics" and therefore require level 2 polarity. This is illustrated in Figure 3.3.

3.4 Construction Of The knowledge base

The knowledge base (KB) stores a set of common verbs and adjectives that are used in common texts and speeches along with their polarity classification. The creation of the KB

is an important part of our approach to the Winograd Schema Challenge. The problems are created using only common words. Therefore, we consider only the verbs and adjectives that can be extracted from English novels, chat groups, news articles and Wikipedia articles. We start from a small set of adjectives which we manually classify and use as the initial KB. Then we use this set along with a set of text corpora to classify more adjectives and verbs and expand the KB. The block diagram of the construction method is illustrated in Figure 3.4.

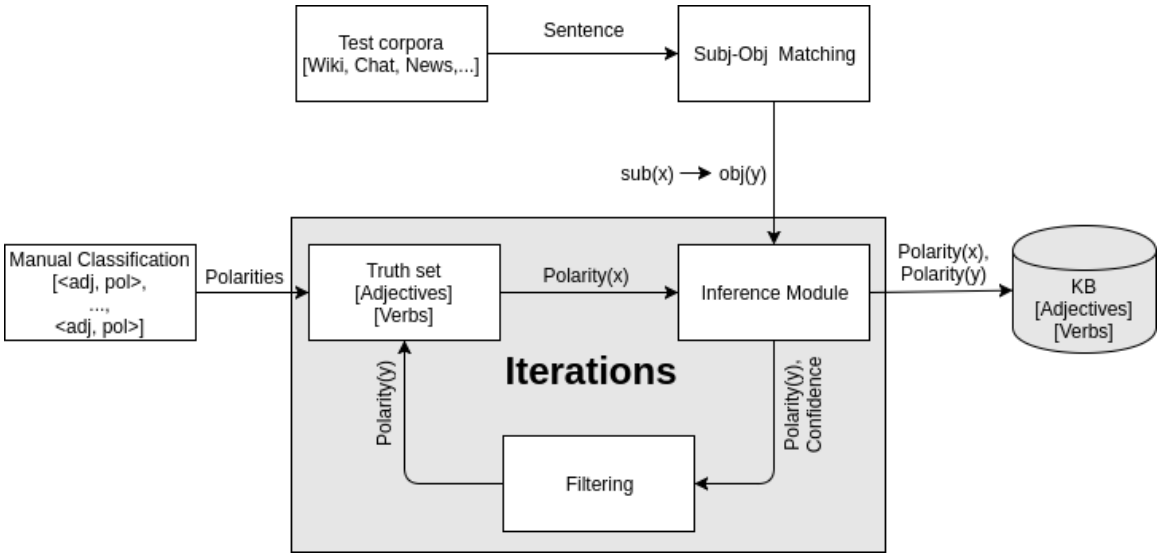


Figure 3.4: Construction of the knowledge base

3.4.1 Description Of Dataset

We created our text corpora using different sources. They are briefly described in the following:

- Wikipedia Abstract: A collection of the abstract sections of Wikipedia articles. This corpus covers the largest number of words.
- NTLK Corpora: The Natural Language Toolkit developed by Loper et al. [22] provides a variety of text corpora. We have used the following resources:
  - Gutenberg Corpus: A collection of $25,000$ electronic books which are hosted at http://www.gutenberg.org/. This corpus represents how adjectives and verbs are used in formal literature.
  - Web And Text Chat: A collection of web texts which includes contents from an online discussion forum, the movie script of “Pirates of the Caribbean”, personal

Corpus	Sentences	Words	Verbs	Adjectives
Wikipedia	7,212,949	9,41,61,710	>20,000	>20,000
Gutenberg	98,552	26,21,785	15,053	9,500
Web Chat	36,074	4,25,024	5,831	5,532
Brown	57,340	11,61,192	11,796	12,364
Reuters	54,716	17,20,917	6,704	4,466

Table 3.3: Description of text corpora

advertisements, and wine reviews. This represents the informal and casual tone of speech.

- Age Specific Chat Group: A corpus of instant messaging chat sessions, originally collected by the Naval Postgraduate School for research on automatic detection of Internet predators. The corpus contains over 10,000 posts, anonymized by replacing usernames with generic names. It is also classified into 15 different age specific groups (e.g., 20s, 30, 40s,...).
- Brown Corpus: The Brown Corpus was the first million-word electronic corpus of English which was created in 1961 at Brown University. This corpus contains texts from 500 sources, and the sources have been categorized by genre (e.g., editorial, news, etc.).
- Reuters Corpus: The Reuters Corpus contains 10,788 news documents. The documents have been classified into 90 topics. Because a news story often covers multiple topics, categories in the Reuters corpus overlap with each other.

A brief comparison of different text corpus is summarized in Table 3.3.

Algorithm 1 Knowledge Base Construction Algorithm

1: **procedure** CREATEKB
Input: Manually Classified Words T, Sentences S, Maximum Iteration Number M
Output: Knowledge Base KB
2: $iter \leftarrow 0$
3: **while** $iter \leq M$ **do**
4: $Votes \leftarrow \emptyset$
5: **for** $x \in T$ **do**
6: $< s, x, y, z, f > \leftarrow ExtractSents(x, S)$
7: $< P, f > \leftarrow GetRelations(s, x, y, z, f)$
8: **if** x is adjective **then**
9: $x[pre][sub] = x[pre][obj] = x[post][sub] = x[post][obj] \leftarrow x[pol]$
10: **for** $< x_{rel}, y_{rel} > \in P$ **do**
11: **if** f is negative **then**
12: Reverse all polarity values of x
13: **if** x is a prefix of y **then**
14: vote in favour of assigning value of $x[post][x_{rel}]$ to $y[pre][y_{rel}]$
15: **else**
16: vote in favour of assigning value of $x[pre][x_{rel}]$ to $y[post][y_{rel}]$
17: **for** any word y who got a vote in current iteration **do**
18: **if** vote of $y \geq \theta$ **then**
19: add y to T
20: $iter \leftarrow iter + 1$
21: $KB \leftarrow T$
22: **return** KB

3.4.2 Using the Dataset to create KB

The next step is to create the knowledge base using the corpora. We start with a small set (the same set of 374 adjectives which was used for constructing the feature vector) of manually classified verbs and adjectives. The corpora is used to expand our initial set of verbs and adjectives in a series of iterations. This is illustrated in Algorithm 1. In the remaining part of the chapter we discuss different parts of the algorithm. Lines 4-20 are executed iteratively. In each iteration, the algorithm performs four tasks:

1. Extraction of Sentences: The task is to extract well formatted sentences from the dataset that can be parsed by our semantic parser. Line 6 of Algorithm 1 performs this task.

2. Obtaining Subject-Object Relations: The task is to parse the sentence to identify the events along with their subjects and objects. Line 7 of Algorithm 1 performs this task.

3. Logical Inference and Voting: The task is to process the relationships of subjects and objects with their respective events using a manually created set of inference rules to determine polarities of the verbs and adjectives of the sentences. Since each verb or adjective can appear in many sentences, each sentence contributes by voting. Lines 8-16 of Algorithm 1 perform this task.

4. Filtering: This step involves counting the votes and comparing them with a predetermined threshold θ to decide the final polarities. Lines 17-19 of Algorithm 1 perform this task.

All of the four tasks are described in detail in the remaining part of this chapter.

3.4.3 Extraction Of Sentences

The size of our text corpora is too big to be used directly. Therefore, once the initial KB is prepared manually, for each word x in the KB, we extract a set of sentences from the corpus. Each extracted sentence s must have the following properties:

- s must contain x and the part of speech tag (POS tag) for x in s must match the POS tag of x in the KB. Some words like "right" can be used both as a noun ("This is my right") or an adjective ("He is the right man for the job"). Therefore, it is necessary to match both the words and the POS tags.

- There should be at least one more verb or adjective y in s such that x and y share at least one common entity z as their subject or object. If x is "brave", s can be: "The brave prince defeated the invaders against all odds" where there is a verb "defeat" such that both "brave" and "defeat" has "the prince" as their common entity. Therefore, y="defeat" and z="prince".

- It is optional to have connectives like "because", "although", "so", "but", or "and" between the clauses of the sentence. If the clause of x and the clause of y is connected by a negative connective, an additional flag is set and sent along with the sentence.

For each sentence s, a tuple is returned in the form of $< s, x, y, z, f >$ where s is the sentence, x, y, and z are the same as mentioned above and f is a flag that determines if x and y are connected by a negative connective. Table 3.6 illustrates some examples.

Event-Event	Event-Entity	Entity-Entity	Entity-Value
objective	agent	is_possessed_by	trait
previous_event	recipient	has_part	size
next_event	object	complement	negative
caused_by	destination/location	is_related_to	dependent
causes	destination	in_conjunction_with	punctuation
inhibited_by	destination/time_at	in_disjunction_with	quantity_modifier
subject	location	identifier	
passive_subject	site/location/time_at	referent	
	beneficiary		
	raw_material		
	origin		
	verb_compliment		
	supporting_verb		
	modal_verb		
	passive_supporting_verb		
	time_between		
	modifier		
	accompanier		

Table 3.4: Relations Of K-Parser

3.4.4 Obtaining the Subject-Object Relations

Once we have a list of sentences in the desired format, our next task is to resolve subjects and objects from the sentences. Subject-Object-Resolution is a popular syntactic ambiguity resolution problem in Natural Language Processing. One of the earliest works on this task was by Macdonald et al. [24]. In recent years, the Stanford Dependency Parser [6] is the most widely used tool for this task which resolves the dependencies using neural networks with very high accuracy. However, in our thesis, we use the parser developed by Sharma et al. [46], known as K-Parser, which is built on top of the Stanford Parser and supports a few additional relationships for the objects. The relationships are categorized by "Event-Event", "Event-Entity", "Entity-Entity", and "Entity-Value" as shown in Table 3.4. It is possible for an event to have multiple objects. The following example can illustrate:

In the sentence: "John poured tea from the kettle to the cup", both "kettle" and "cup" are objects. Although they are treated equally in Stanford Parser, K-Parser supports relationships like "origin" and "destination" to distinguish them. Moreover, it also supports all functionalities of the Stanford Parser. Although in this thesis, we do not consider different types of objects, we use K-Parser to consider possible future extensions of our approach.

All the sentences that are extracted are parsed to determine the subject-object relationship between x and z, and between y and z. It is possible that z may not be directly related to either x or y. In that case, if any coreference of z is related to x or y, the relationship is determined based on the coreference. The sentences in standard English texts are significantly less ambiguous than the Winograd Schema and therefore, K-Parser works reasonably well for the coreference resolution. For subject object matching, we use the relations of K-Parser to determine whether z or its coreference is a subject or object of x and y.

Each sentence s goes through preprocessing, subject-object resolution and connective processing. We discuss all the steps briefly in the following.

Preprocessing

Unstructured sentences need to be preprocessed in order to be parsed properly. We use a few hand generated rules for the preprocessing step.

- Contraction Removal: Some of the examples of contractions are "can't" (can not), "won't" (will not), "should've" (should have), etc. These contractions contain information which can not be used in contracted form. Because of the irregular pattern of the contractions, we use a manually created dictionary to replace contracted words by the expansions.

- Processing Modifier: Modifiers can be either positive or negative. Negative modifiers (e.g., "less", "hardly") have the ability to change the meaning of the sentences. Positive modifiers (e.g., "very", "more") usually emphasize but not change any meaning. For any negative modifier, we replace it by "not". For any positive modifier, we simply remove it.

- Unification Of Entities: Sometimes, two or more entities appear in a sentence in a possessive relationship. this type of entities can be merged together without loss of information. For example "Joe's uncle" or "Ann's son" refers to only one entity and can be merged together to form "Joe's_uncle" and "Ann's_son".

Subject Object Resolution

For K-Parser, x and y are marked either as events (if verbs), or traits (if adjective). For both x and y we find the chain of relationships with z using Table 3.4. In most of the cases, x and y are directly related to z. However, in some cases, the distance can be more than 1 step. If either x or y is event, we determine whether z is the subject and the object based on the relationship between the event and the root of the subtree which contains z. If the relationship is "agent", then z must be the subject. If the relationship is anything other than "agent", then z must be the object. On the other hand, if either x or y is a trait, then z must be the subject.

Connective Processing

K-Parser converts the connectives into relations between events. The relation can be either "causes", "caused by", "next event", or "previous event". The temporal ordering is represented by an arrow from the previous event to the next event. We place the arrow so that the arrowhead points from the previous event to the next event or from the cause to the effect.

The temporal ordering of x and y is determined using the following rule:

- If x and y are connected by "though", "although", "after", or "because", whatever appears before the connectives is positioned later in the temporal ordering. For example, in the sentence: "Alex passed because he is hardworking", "hardworking" appears before "pass" in temporal ordering.
- If x and y are connected by "so", "but", "and", or "before", whatever appears after the connectives is positioned later in the temporal ordering. For example, in the sentence: "Alex passed but he was rejected", "reject" appears after "pass" in temporal ordering.
- If x and y are not connected, the ordering of words in the sentence is maintained in the temporal ordering as well.

K-Parser does not distinguish between positive and negative events and therefore "although" and "because" are treated similarly, which can lead to incorrect parsing. Therefore, we use the flag f to check if the events are connected with a negative connective. Table 3.5 illustrates how connectives are processed.

Sentence	Sentence(Modified)	K-Parser Output	Final Output
C1 because **C2**	-	**C2** causes **C1**	**C2→C1**
C1 so **C2**	-	**C1** causes **C2**	**C1→C2**
C1 and **C2**	-	**C1** next_event **C2**	**C1→C2**
C1 but **C2**	**not C1** and **C2** **C1** and **not C2**	**not C1** next_event **C2** **C1** next_event **not C2**	**not C1→C2** **C1→not C2**
C1 although **C2**	**not C1** because **C2** **C1** because **not C2**	**C2** causes **not C1** **not C2** causes **C1**	**not C2→C1** **C2→not C1**

Table 3.5: Connective Processing

For each sentence s with connected events x and y, a tuple is returned in the form of $< p, f >$ where p is the subject-object pair which is sorted according to the temporal ordering and f is the flag that indicates whether x and y are connected by a negative connective. The value of f is "true" if the connective binding x to y is negative. Table 3.6 illustrates some examples.

Sentence	x	y	z	f	p
The prince defeated the invaders because he is brave	brave	defeat	the prince	false	sub(brave)→sub(defeat)
The prince was defeated by the invaders because he is weak	weak	defeat	the prince	false	sub(weak)→obj(defeat)
The prince was defeated by the invaders and he became weak	weak	defeat	the prince	false	obj(defeat)→sub(weak)
The prince was defeated by the invaders but he was not weak	not weak	defeat	the prince	true	obj(defeat)→not sub(not weak)

Table 3.6: Subject Object Resolution

3.4.5 Logical Inference and Voting

The inference module receives the polarity of a verb or adjective x from the KB along with the subject-object relations and flags. If x is an adjective, it has only one polarity having only one feature with two possible values (negative/positive) for level 1 polarity and two features with four possible values for level 2 polarity (ability: negative/positive, ethics: negative/positive). However, if x is a verb, it has four different polarities as shown in Figure 3.4. Therefore, it can have 16 possible values for level 1 polarity and 256 possible values for level 2 polarity. These values indicate the prior and posterior polarity of its subject and object. All the values of x are available from the initial manual classification.

In order to treat verbs and adjectives equally, we use the same data structure for adjectives. If x is an adjective, we treat any entity which is associated with x as both the subject and object of x. For the sentence, "Alex is brave", we consider Alex as both the subject and object of "brave". Our assumption is that any word for which an adjective is applied to (e.g., "brave" is applied to "Alex") has same polarity as the adjective itself, and acts as both the subject and object before or after the adjective. Since "brave" has positive polarity, the "pre" and "post" polarity of the subject and object of "brave" (in this example: Alex) should also have positive polarity, because adjectives are static. This assumption enables us to process both adjectives and verbs similarly using same data structure.

After sentence extraction, we get a tuple in the form of $< s, x, y, z, f >$ for each sentence. For each relationship pair of x and y, we compute the polarities of y using the following rules:

- If y appears before x in temporal ordering, the pre polarity values of x are used to determine the post polarity values of y.
- If y appears after x in temporal ordering, the post polarity values of x are used to determine the pre polarity values of y.
- In both cases, if the flag f is set, the polarity is reversed.

Because a single word x can appears in multiple sentences with multiple y, from each word in KB, we can determine polarity of multiple new words. Therefore, this procedure is repeated for all the words in KB. At the end of an iteration, all the y which are not members of the KB are listed along with their votes.

Table 3.7 illustrates the process of inference and voting. In KB, all the words are stored in the format: $< Pol(Sub)_{pre}, Pol(Obj)_{pre}, Pol(Sub)_{post}, Pol(Obj)_{post} >$. We assume that the words "brave" and "weak" are already in KB and y="defeat", which is a new word. Therefore, at the beginning, "defeat" has the polarity $<?,?,?,?>$ where "?" means that the polarity has not been determined. After encountering a sentence involving "defeat", the logical inference is performed and the polarity of y is updated. In the last column, the polarity of y is shown after encountering each example e. The votes are shown as superscripts on top of each polarity value.

e	p	x (From KB)	f	y (New)
1	sub(brave)→sub(defeat)	$brave =< 1,1,1,1 >$	false	$defeat =< 1^1,?,?,? >$
2	sub(weak)→obj(defeat)	$weak =< 0,0,0,0 >$	false	$defeat =< 1^1,0^1,?,? >$
3	obj(defeat)→sub(weak)	$weak =< 0,0,0,0 >$	false	$defeat =< 1^1,0^1,?,0^1 >$
4	obj(defeat)→not sub(not weak)	$not\,weak =< 1,1,1,1 >$	true	$defeat =< 1^1,0^1,?,0^2 >$

Table 3.7: Logical Inference and Voting

3.4.6 Filtering and Expansion

After processing each word x in KB and getting their votes, we end up with a list of new words along with their votes. For each new word y, if the vote is higher than a specific value θ, the word is added to KB and this KB is used in the next iteration. We select the value of $\theta = 5$. Therefore, the verbs and adjectives whose polarity value is not supported by at least 5 sentences are filtered out. After the filtering step, the new words along with their polarities are added to the KB. In this way, the KB is expanded in each iteration. By controlling the number of iterations, we can control the size of KB. Table 3.8 shows the content of KB after each iteration. We also measure how many problems from the Winograd

Schema set ($WSCL$) and Rahman's set ($WSCR$) are covered by the verbs and adjectives of KB. After two iterations, most of the problems in both problem sets are covered.

Iter	Adj	Verb	Over-lap	WSCL Covered (Total: 285)	WSCR Covered (Total: 564)
0	374	0	0	0	96
1	11408	5484	3149	272	528
2	21842	11098	7907	279	549

Table 3.8: Knowledge Base at each Iteration

From Table 3.8, it is also evident that each iteration adds more new words to the knowledge base (KB). However, the extended KB does not cover more problems in any of the two sets. As the size of KB grows, the iteration becomes much more time consuming with a negligible increase in coverage for both the problem sets. Therefore, after iteration 2, we stop expanding.

Although, KB contains a large number of verbs and adjectives, we can not validate the accuracy because the structure of KB is different from existing polarity knowledge bases like [55] or [25]. None of the existing knowledge bases consider dynamic polarity for verbs. Moreover, since our method for determining polarity is based on both inference rules and statistics (voting), the possibility for errors gets higher with further iterations. This is another reason of not expanding KB beyond iteration 2.

Chapter 4

Methodology: Solving the Winograd Schema

In chapter 3, we discussed how we created a knowledge base to store the polarity of common verbs and adjectives. In this chapter, we discuss how we use the knowledge base to solve the Winograd Schema Challenge.

4.1 Identifying the Solvable Problems

Not all types of Winograd Schema problems are solvable by our approach. In order to be solvable by our approach, a problem must have the following properties:

- Both the pronoun and the candidate referents must be associated with at least one event or adjective.
- All events must contain at least one verb which denotes the action. Having either subject or object is optional.
- The information required to resolve the coreference must be logically obtainable from the events, adjective, and the relationships which bind the pronoun and the candidate referents with their events or adjectives.

If we consider the following problem: "There is a shark swimming below the duck. It should get away to safety fast", we note that the information to resolve the pronoun does not exist in events like "swim" or "get away" or their relationships with their entities. In order to solve the problem, we need to analyze the property of "shark" and "duck". If we replace "shark" and "duck" by x and y, the problem becomes unsolvable. Our approach can not solve problems like this. However, if the problem is modified to: "There is a ferocious shark swimming below the duck. It should get away to safety fast", the adjective "ferocious" acts as a trait for the word "shark". In this case, if we replace "shark" and "duck" by x and y, the problem is still solvable. This is solvable by our approach.

We sampled a set of 133 problems from the Winograd Schema problem set $WSCL$, which satisfy the properties mentioned above. In addition to that, we have also found that all 564 problems from the test set of Rahman et al.'s [39] problem set $WSCR$ satisfy the above three criteria but are not necessarily google-proof. All our experiments are performed on these 2 sets. In order to evaluate the performance of our approach, we compare it with the performance of a baseline solution on the same problem sets.

4.2 Creating a Baseline

We create the baseline based on Peng's approach [37] described in chapter 2. However, Peng et al. tested the accuracy of their approach on $WSCR$, which is easier for statistical approaches compared to $WSCL$ because of having redundant information. We create the baseline using a few simple modifications on Peng's approach to make it compatible to $WSCL$ which are described in the following:

- Redundancy Removal: Peng described two types of predicate schema:
  - Type 1: Given a predicate $pred$, a mention m which is the subject of $pred$, and any other non-subject mention a associated to $pred$, Type 1 schema has the form $< pred(m, a) >$, which determines the probability of finding $pred$, m, and a in a large corpus with same mutual relationship.
  - Type 2: Given two predicates $pred$ and $\widehat{pred}$, their subject m and $\widehat{m}$, object a and $\widehat{a}$, and the connective cn, Type 2 predicate has the form $< pred(m, a)|\widehat{pred}(\widehat{m}, \widehat{a}), cn >$, which determines the probability of finding the predicates in a large corpus where their subjects and objects maintain same mutual relationship.

  The authors proposed a method to calculate the scores of each schema that are present in the problem set based on a vector of four different scoring functions. The score S is a vector of $S_{giga}, S_{wiki}, S_{web}$ and S_{pol}, where the "giga", "wiki", and "web" scores are calculated based on the frequency of the schema in Gigaword corpus, Wikipedia corpus, and Google web query. However, among the four scoring functions, only S_{giga} supports both version of predicate schema whereas other three functions support only Type 1 schema. For the Winograd Schema, Type 1 schema is not very useful and often misguiding which leads to incorrect answer because of the google-proof property. Therefore, we only use S_{giga} and remove other three redundant scoring function.

- Probability Calculation: All Winograd Schema problems have two candidate mentions and replacing the pronoun by each of them results in two alternative answers. We calculate the joint probability of the likelihood of each alternative. For the sentence: "The fish ate the worm because it was hungry", the alternatives are:

A_1 =The fish ate the worm because the fish was hungry.
A_2 =The fish ate the worm because the worm was hungry.
We can calculate the joint probability of A_1 as follows:

$$\begin{aligned} P(A_1) &= P(The\ fish\ ate\ the\ worm\ because\ the\ fish\ was\ hungry) \\ &= P(The\ fish\ was\ hungry) \times P(The\ fish\ ate\ the\ worm \mid The\ fish\ was\ hungry) \\ &= P(Hungry(fish, *)) \times P(Eat(fish, worm)|Hungry(fish, *)) \end{aligned} \tag{4.1}$$

From the right side of Equation 4.1, the first term is same as Peng's scoring function S_{giga} for Type 1 schema and the second term is similar to the scoring function of Type 2 schema. However, Peng used the logarithmic count. Taking the log for both side of 4.1, we get:

$$\begin{aligned} Log(P(A_1)) &= Log(P(Hungry(fish, *))) + Log(P(Eat(fish, worm)|Hungry(fish, *))) \\ &= S_1(Hungry(fish, *)) + S_2(Eat(fish, worm)|Hungry(fish, *), because) \end{aligned} \tag{4.2}$$

Similarly:

$$Log(P(A_2)) = S_1(Hungry(worm, *)) + S_2(Eat(fish, worm)|Hungry(worm, *), because) \tag{4.3}$$

Therefore, according to Equation 4.3, the logarithmic probability for any alternative is the sum of Type 1 and Type 2 scoring function.

- Dataset Selection: Peng et al. used Gigaword corpus, which is different from our corpus. To ensure fair comparison, we calculate all probability scores based on the corpus mentioned in section 3.4.1.

- Sparsity Reduction: Peng's Approach requires exact word matching for counting the scores. It is often difficult to find such a match, specially if the data set is relatively small. We reduce the sparcity by replacing each word by their Wordnet Synset and aggregating the count for all possible combinations of the synonyms.

- ConceptNet Addition: For Type 2 schema, there are some cases where it is difficult to find words or triples that maintain exact same relationship as the Winograd Schema problem because the problems are designed to contain implied relationship. For example, in the sentence "The ball crashed through the table because it was made of steel", the relationship between "made of steel" and "crash" is not obvious. However, The relationship between "steel" and "hard" and relationship between "hard" and "crash" is very common in Wikipedia. The word "hard" acts as a bridge to connect "steel"

and "crash". This type of intermediate relationships can be learned from ConceptNet. We enhanced the baseline by allowing the algorithm to search for a path between the words. To avoid time out, we restrict the path to have at most 2 edges. In any of the paths, if we find an intermediate word $pred_w$ between $pred$ and $\widehat{pred}$, we calculate the probability as the following:

$$P(pred(m,a)|\widehat{pred}(\widehat{m},\widehat{a})) = P(pred(m,a)|pred_w(*,*)) \times P(pred_w(*,*)|\widehat{pred}(\widehat{m},\widehat{a})) \tag{4.4}$$

Where the asterisk indicates that any word is allowed to form triple with $pred_w$. The probabilities can be easily converted to the logarithmic form similar to 4.3.

The above modifications make the baseline suitable for the original Winograd Schema set $WSCL$. We apply the baseline on both $WSCL$ and $WSCR$ and compare its performance with our approach using the knowledge base which is discussed in detail in the following section.

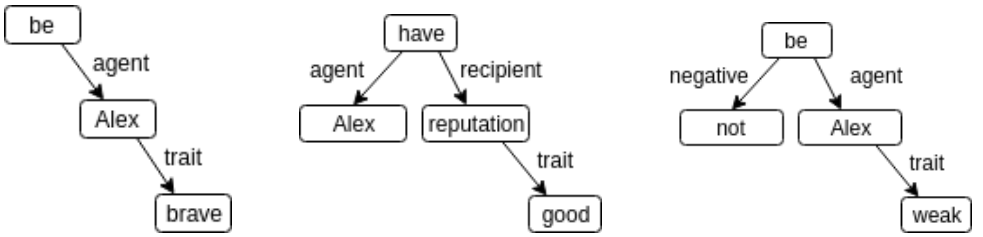
4.3 Creating a Solution Using the Knowledge Base

The Winograd Schema problems are expressed in unstructured natural language. We use the same parser (modified from K-Parser [46]) that was used to create the knowledge base (KB) from text corpora. Unlike the sentences from the text corpora which were used to create the knowledge base, the Winograd Schema sentences are more ambiguous and coreference resolution is much harder. Therefore, we create alternative copies of the text by replacing the pronoun with each of the candidate mentions.

When using the parser on the Winograd Schema text, one of the most important tasks is to identify all the events in the text. The K-Parser created by Sharma et al. [46] provides support for single-word events only. In the text: "He is moving out of his old house", the event identified by the K-Parser is "move". This property was useful when we created the KB because we never stored any multy-word event in KB. However, for the Winograd Schema problems, we need to consider multy-word events. For example, if we use the single word "move" as the event, the polarity of the subject and object of "move out" will be the same as the polarity of the subject and object of "move in", which will lead to many inaccuracies for answering the Winograd Schema problems where the twin sentences are differed by only a single word. To avoid this scenario, we add a functionality to the K-Parser to create multy-word events by concatenating the single-word event with associated modifiers, which allows our approach to distinguish between "move out" and "move in". Moreover, we have developed another functionality to derive the polarity of the multy-word event from the polarities of its words.

For each Winograd Schema problem, our approach consists of three tasks.

- Formation of arbitrary length events.



(a) Presence of "be" event (b) Presence of "have" event (c) Presence of "negative" relationship

Figure 4.1: Construction of events from K-Parser output

- Determination of polarity for arbitrary length events from its words.
- Determination of polarity for pronoun and candidate referents and solving the Winograd Schema.

We discuss all three tasks in the following sections.

4.3.1 Event Identification

The events consist of one or more words that refer to an action and usually have an agent (subject) and one or more recipients (object). We use the following steps to process the K-Parser output to form the events.

- Redundancy Removal: All "identifier" relations are removed to make the parse tree concise.
- Identifying Event Root: We start with each of the candidate mentions and the pronoun of the Winograd Schema sentence and find their ancestors in the parse tree. We keep going up the tree until we find a word that can be classified as "Event". To select an event root, we consider three possible cases. All the cases are illustrated in Figure 4.1
  - "Be" Event: Consider the sentence, "Alex is brave". Words like "is", "was", "are", etc. are always considered as "be" event in K-Parser whereas the adjectives are considered as "traits". However, the adjective carries significant information and therefore, in our approach, we want the adjective to be included in the event. In the case of "be" event, we take the "trait" as the root for the event. This is illustrated in Figure 4.1(a).
  - "Have" Event: Consider the sentence, "Alex has a good reputation". In this case, the adjective is associated as a "trait" to a secondary word "reputation", which is again connected to a "have" event as "recipient". Therefore, in the case of "have" we take the "trait" of the "recipient" as the root event. This is illustrated in Figure 4.1(b).

- Other: If the event is anything other than "be" or "have", we accept the event as the root.

- Adding Negation: Consider the sentence, "Alex is not weak". Here, the event "be" has a negative relationship. If the event has such a "negative" relationship, we add "not" at the beginning of the root. This is illustrated in Figure 4.1(c).

- Adding Complementary Words: Complementary words are the words which are associated with the root event by "raw material", "modifier", "object", "dependent", "verb complement", or "supporting verb" relationship. Complementary words are added at the beginning. In the presence of negative events, the modifiers are combined with "not" to form negative modifiers like "can not", "do not", "will not", etc.

After the events are formed, the subject object resolution and the connective processing is performed as described in section 3.4.4. After that, we have two remaining tasks: determining the polarities of events which may contain multiple words, and determining the polarities of the entities. These tasks are very similar to Ma et al.'s [23] approach. However, as we discussed in chapter 2, we can not use a supervised method like Ma et al. because of the lack of suitable training data. Therefore, in this thesis, we adopt a relatively simpler rule-based approach that can calculate polarity of events from the constituent words without any training. It remains as a future work to create a training set for aspect based polarity analysis using our polarity features which can enable us to use supervised polarity determination techniques for the Winograd Schema problems.

4.3.2 Polarity of Multi-Word Events

After adding complementary words and negation, the events may contain multiple words. From the following example: "I can not cut the tree down with the axe", the event is extracted as "can not cut down". The polarity determination is performed in two steps: i) determination of polarity without considering negation and ii) handling negation. The complete flowchart is illustrated in Figure 4.2.

Determination of Polarity Without Considering Negation

In this step, we determine the polarity by ignoring negative words. The negative words are "not", "less", "without" etc. Therefore, the event "can not cut down" becomes simply "cut down". Based on the semantic meaning, we categorize all the multy-word events in four categories.

- Verb Dominant: This is the most common category. As we already discussed, all events must contain at least one verb. In most of the cases, the semantic meaning of the verb determines the semantic meaning of the event and therefore the polarity of the event

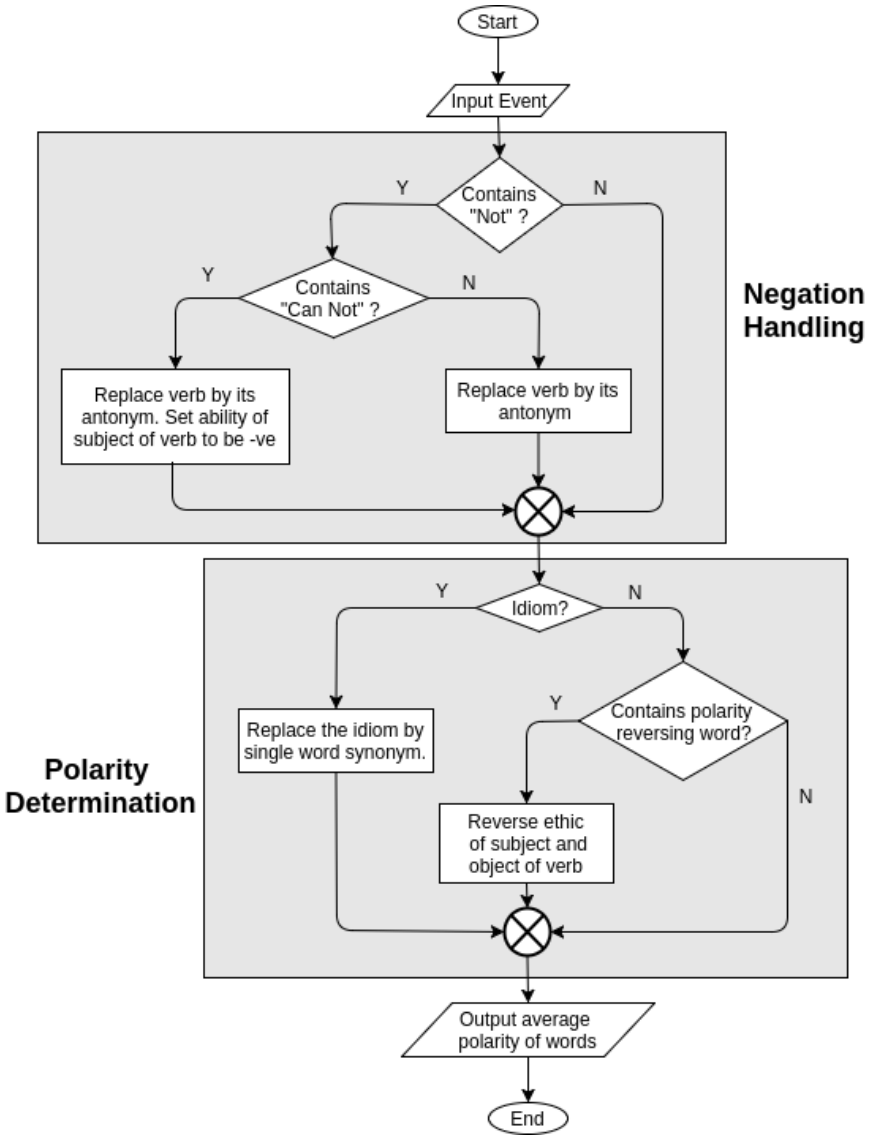


Figure 4.2: Flowchart of multy-word Polarity Determination

becomes same as the polarity of the verb. For example, the polarity of the subject and object of "cut down" is same as the polarity of the subject and object of "cut".

- Adjective/Trait Dominant: Sometimes the event contains adjectives and the polarity of the adjective determines the polarity of the event. For example, from the sentence "The bus was going slow", the extracted event is "go slow". Here, the word "go" has neutral polarity in terms of both ability and ethics. However, the trait "slow" determines that the ability of the subject of "go slow" should be negative. This usually occurs when the verb has neutral polarity.

- Reverse Verb Dominant: Some words are capable of reversing the polarity of the verb of an event. Examples of such words are: "stop", "against", "hate", "hinder", etc. for

example: polarity of the subject of "hate to lie" is positive although the polarity of the subject of "lie" is negative. It is important to note that the word "hate" is not a negation and therefore can not be omitted beforehand.

- Idioms: These are special events where two or more words express entirely different semantics than the words themselves. For example: "calling a day" represents "stopping" instead of "calling" or "day".

In order to determine polarity, we start with resolving the idioms by searching the word sequence in the Merriam Webster dictionary [26]. If the word sequence is found, we search through the synonyms to find any synonym that consists of a single word. The synonym is used as a replacement to the event. If the word sequence is not found in the dictionary, we assume that it is not an idiom.

The next step is to search for the words capable of reversing the polarities of verbs. We use a hand made set of words for this purpose. If we find such word, the polarity of the verb is reversed. For example: "lie" has a negative polarity for level 1 and a negative "ethic" in level 2. Therefore, "hate to lie" reverses the value to be positive.

In the last step, we split the sequence into a list of words and perform a majority voting among the words to determine the polarity of the event. This technique works because in most of the cases, only one word in the event has non-neutral polarity. In the verb-dominant events, any word other than the verb usually has neutral polarity (e.g., "fit in", "throw down", "crash through") whereas in most of the adjective dominant events, any word other than the adjective usually has neutral polarity (e.g., "arrive later", "go slow", "become bold"). There are some exceptions where the votes may contradict with each other (e.g., "offer insult", "lose soundly", "cede to popular"). In these cases, the vote of the verb gets priority.

Negation Handling

After the polarity of the events without negation is determined, we modify the polarity by handling negation. For this task, our objective is to determine how negative words influence the polarity of an event. There are a large number of existing approaches which can handle negation. However, none of them are compatible with our polarity structure. Therefore, we use a simple set of rules to modify polarities of events based on negative words. First, we analyze the presence of negative word (i.e., "can not", "is not", etc.). If "can not" is present in the event, we set the value of "ability" as negative. For any other negative word, we replace the verb of the event by its antonym according to the Merriam Webster dictionary. Table 4.1 shows a sample of multy-word events along with their polarities determined using our method. A(s), E(s), A(o) and E(o) denotes the ability and ethics of subject and object respectively. "Pre" and "Post" denotes to the "pre" and "post" polarity values.

Word	A(s) (pre)	A(s) (post)	E(s) (pre)	E(s) (post)	A(o) (pre)	A(o) (post)	E(o) (pre)	E(o) (post)	Comment
Influence	pos	pos	-	-	-	-	-	-	Verb
Partially	-	-	-	-	-	-	-	-	Adj
Influence Partially	pos	pos	-	-	-	-	-	-	Verb Dominant
Zoom	pos	pos	-	-	neg	neg	-	-	Verb
By	-	-	-	-	-	-	-	-	Modifier
Zoom By	pos	pos	-	-	neg	neg	-	-	Verb Dominant
Go	-	-	-	-	-	-	-	-	Verb
Slow	neg	neg	-	-	-	-	-	-	Adj
Go Slow	neg	neg	-	-	-	-	-	-	Trait/Adj Dominant
Fear	neg	neg	-	-	pos	pos	-	-	Reverse Word
Violence	-	-	neg	neg	-	-	-	-	Verb
Fear Violence	neg	neg	pos	pos	-	-	-	-	Reverse Verb Dominant
Sleep	-	-	-	-	-	-	-	-	Verb
Sleep With	-	-	neg	neg	-	-	-	-	Idiom

Table 4.1: Different Types of multy-word Events

4.3.3 Polarity of Entities and Solving the Winograd Schema

Each Winograd Schema problem contains a pronoun and two candidate referents. The parser recognizes the pronouns and referents as entities. Each entity is associated with an event. After calculating the polarities of all events, the polarity of the entities are determined by their associated events and their relationships (subject or object) to the events. Polarity of the event contains the information about how the polarities of the subject and object should be. For example, the polarity of "zoom by" event indicates that the subject should have positive "ability" and neutral "ethics" whereas the object should have negative "ability" and "neutral" ethics. This is illustrated in Table 4.2.

Entity	Event	Relationship	A_{pre}	A_{post}	E_{pre}	E_{post}
The Truck	zoom by	S	pos	pos	-	-
The Bus	zoom by	O	neg	neg	-	-
It	go slow	S	neg	neg	-	-

Table 4.2: The Truck zoomed by the Bus because it was going slow

The connectives play a crucial role in resolving the coreference for dynamic polarity events. The connectives determine whether the "pre" or "post" polarities of the entities should be used. In the sentence: "The truck zoomed by the bus because it was going slow", we have two sub-sentences joined by a connective "because". If we express the sub-sentences by **C1** and **C2** where **C1**="The truck zoomed by the bus" and **C2**="It was going slow", the sentence becomes: "**C1** because **C2**". According to Table 3.5, the parser represents this sentence as: **C2** $\rightarrow$ **C1**. If the connective is changed from "because" to "so", the representation of the sentence becomes: **C1** $\rightarrow$ **C2**. If the entity within **C1** is X_1 and the entity within **C2** is X_2, in order to be co referents, X_1 and X_2 must satisfy the following conditions:

If **C1** $\rightarrow$ **C2** (e.g. **C1** so **C2**, **C1** and **C2**, **C1** before **C2**) :

$$Pol(X_1)_{post} = Pol(X_2)_{pre} \tag{4.5}$$

If **C2** $\rightarrow$ **C1** (e.g. **C1** because **C2**, **C1** after **C2**) :

$$Pol(X_1)_{pre} = Pol(X_2)_{post} \tag{4.6}$$

Else :

$$\begin{aligned} Pol(X_1)_{pre} &= Pol(X_2)_{pre} \\ Pol(X_1)_{post} &= Pol(X_2)_{post} \end{aligned} \tag{4.7}$$

In our example, the connective is "because". Therefore, if we consider X_1 to be "the bus" and X_2 to be "it", then from Table 4.2, $A(\text{the bus})_{pre} = A(it)_{post} = neg$ and $E(\text{the bus})_{pre} = E(it)_{post} = neutral$. Therefore, "it" refers to "the bus". On the other hand, if we consider X_1 to be "the truck", we see that $A(\text{the truck})_{pre} = pos \neq A(it)_{post}$ and therefore, "it" does not refer to "the truck".

4.4 Experimental Results

In this section, we present the results obtained from comparing the effectiveness of our approach. *WSCL* [19] is the official Winograd Schema problem set created by Levesque. There are a few other problem sets (e.g., PDP [9], WSCR [39], SuperGLUE-WSC [52],

KnowRef [13], Winogender [41]) that are inspired by $WSCL$. However, since most of the previous logical and statistical approaches have used $WSCL$ and $WSCR$, we select these datasets for evaluation. For $WSCR$, we only consider the 564 problems from the test set because none of the previous approaches were judged based on the training set. For the Winograd Schema set $WSCL$, we also compare the performances of all previous approaches and human participants.

4.4.1 Experimental Setup

We discussed in section 4.1 that all forms of the Winograd Schema are not solvable by our approach and presented three conditions that must be satisfied in order to be solvable by our approach. The size of the problem set solvable from $WSCL$ is 133. For $WSCR$, all 564 problem instances satisfy the conditions and therefore solvable by our approach. $WSCR$ is criticized by Sharma et al. [46] for violating the google-proof property of the Winograd Schema. Therefore, from the 564 problems, we manually select a smaller set of problems which are verified as google-proof. This selection is carried out by replacing the target pronoun by both candidate referents and checking if the number of responses from Google with one option is significantly larger than the other. If such a difference is present, we reject the problem. We found 79 problems out of 564 which are not google-proof. The remaining 485 problems create a third problem set $WSCRGP$ which we also use for validating the performances of both our methods and the baseline.

Our approach produces four methods. SW indicates that only single-word events are allowed whereas MW considers both single-word and multy-word events and calculates the polarity according to Figure 4.2. L1 and L2 denotes Level 1 and Level 2 polarities respectively. Leve1 1 polarity considers a single value for polarity whereas Level 2 considers "ability" and "ethics". The answers are evaluated as either right or wrong. Our approach covers a particular subset instead of the full problem set. Therefore, we also take the coverage into account. If the solution process is interrupted before the reasoning step, we consider the problem as "not covered" by the particular method. This situation arises if the parser fails to parse the sentence and can not identify the relationships of the pronouns and candidate referents properly. For the baseline, "not covered" indicates a scenario where the method fails to find triples for predicate schema 1 and predicate schema 2 and returns a zero count for both of them. The baseline, which was created based on Peng's [37] statistical approach, can not provide an answer if the schema count for both Type 1 and Type 2 are zero. In order to take the coverage into account, we consider two types of accuracy:

- Relative Accuracy: The ratio of correct answer to the problems that are covered. For our methods, it is a measurement of the accuracy of the logical inference module.
- Accuracy: The ratio of correct answer to all problems in problem set (133 for $WSCL$ and 564 for $WSCR$). It measures the accuracy of the overall system.

4.4.2 The Winograd Schema Set (WSCL)

For the Winograd Schema set, we compare all four methods of our approach with all previous statistical and logical approaches and also with human performance. The accuracy of Sakaguchi, He, Kocijan, Trinh, Opitz, Emami, Sharma, Liu, and Human performance are directly taken from the published articles. It is important to note that the methods of Sakaguchi, He, Kocijan, Trinh, Opitz, Emami, and Liu were applied to a larger set of problems from $WSCL$ compared to our approaches. For these methods, we consider both the "accuracy" and "relative accuracy" as equal to the reported accuracy. For existing works which proposed multiple methods, we consider the best of them in terms of accuracy over the Winograd Schema set (e.g., "Ensemble of 14 LMs" for Trinh et al. and "Siamese Ensemble" for Opitz et al.). For Emami et al., we consider the method with random guessing because it can solve 37 more problems correctly compared to the method without guessing. It also has the highest reported accuracy among all of their proposed methods. The results are illustrated in Table 4.3.

Approach	Correct	Coverage	Problem Set	Relative Accuracy	Accuracy
Sakaguchi	-	273	273	90.10	90.10
He	-	273	273	75.10	75.10
Kocijan	-	273	273	72.50	72.50
Trinh	-	273	273	63.70	63.70
Opitz	-	282	282	54.00	54.00
Emami	119+37	273	273	57.14	57.14
Sharma	49	71	-	69.01	-
Liu	-	273	273	52.80	52.80
Baseline	65	88	133	73.86	48.87
SW-L1	66	127	133	51.97	49.62
MW-L1	78	127	133	61.42	58.65
SW-L2	81	127	133	63.78	60.90
MW-L2	97	127	133	76.38	72.93
Human	-	160	160	92.10	92.10

Table 4.3: Performance Comparison: The Winograd Schema (WSCL)

L1 vs L2 Polarity

From Table 4.3, we can see that L2 polarity based method has significantly higher accuracy than L1 polarity based method for both single-word and multy-word events. This result shows that treating polarity as a vector with multiple features increases the accuracy for the reasoning module. Considering "ability" and "ethics" as independent features allows

our reasoning module to align the pronoun with the correct candidate for many problem instances where the traditional atomic polarity based reasoning fails.

Single-Word vs Multi-Word Events

Table 4.3 shows that considering multy-word Events has positive effect on the accuracy. If we only consider single-word events, we have no choice but to determine the polarity of all events based on its root verb or adjective without considering the modifiers and auxiliary words, which leads to lower accuracy. Our results also indicate that the performance gap between single-word and multy-word based methods increases with polarity level. For L1 polarity, using multy-word events causes the accuracy to increase from 49.62% to 58.65% whereas for L2 polarity, it increases from 60.90% to 72.93%.

Our Approaches vs Baseline

From Table 4.3, we can see that the relative performance of the baseline (73.86%) is higher than our methods except MW-L2. This result is interesting because the baseline is based on Peng's approach which is designed to exploit the property of non google-proof problems to compare number of triples for Type 1 and Type 2 predicate schema. This property of Peng's approach is ineffective in handling the original Winograd Schema. However, our modifications of Peng's method is more suitable for the Winograd Schema problems and therefore the accuracy of the baseline is improved.

Even though the relative accuracy is higher, the final accuracy of the baseline is lower than all of our methods because of its low coverage. This scenario can be explained by the observation that for any type of predicate schema, in order to gather knowledge from a commonsense sentence, all the words of the predicate schema must also be present in the commonsense sentence. For example, we can consider the problem: "The bee sat on the flower because it has pollen". Type 1 predicate schema for this problem is: "has(bee, pollen)" and "has(flower, pollen)", which requires commonsense sentences that contain both "bee" and "pollen" and both "flower" and "pollen". This type of sentence is easily available on large text repositories. However, in our experiment, our text repository is significantly smaller than the one used by Sharma et al. and Peng et al. Moreover, the Winograd Schema set is designed to make this type of searching ineffective. Therefore, in the Winograd Schema set, the baseline suffers from lower coverage which results in lower final accuracy. To compensate this, the baseline needs a much larger commonsense data set than what is needed for our methods to achieve similar performance.

However, it is interesting to note that our most effective method MW-L2 outperforms the baseline in terms of both relative and final accuracy. This proves the effectiveness of our approach over the baseline in Winograd Schema set.

Our Approaches vs Previous Logic Based Approaches

Sharma's approach covers 71 problems. Among them, 49 are answered correctly, 4 incorrectly and remaining are left unanswered with an accuracy of 69.01%. Emami's approach covers 273 Winograd Schema because it uses a relatively simple and flexible parser. The increased coverage comes with a drawback of lower accuracy of 57.14%. MW-L2 outperforms Sharma's approach both in coverage and accuracy. MW-L2 has a lower coverage than Emami's approach. However, the accuracy of SW-L2, MW-L1 and MW-L2 are higher than Emami's approach.

Our Approaches vs Previous Statistical Approaches

Statistical approaches are applicable for all types of Winograd Schema problems. Therefore, all the statistical approaches have much higher coverage than our approach. However, in terms of accuracy, only Kocijan et al.'s, He et al.'s and Sakaguchi et al.'s approaches show similar accuracy to MW-L2. All three approaches that outperform MW-L2 use very large training sets and transfer learning. It is also important that the relative accuracy of MW-L2 is higher than Kocijan et al. However, because of relatively lower coverage, the final accuracy of MW-L2 remains almost similar to Kocijan et al.

The accuracy of SW-L1, SW-L2, MW-L1 and MW-L2 are significantly lower than He et al. and Sakaguchi et al. Therefore, it is clear that, our approach can not outperform the state of the art statistical approaches.

4.4.3 Rahman's Problem Set (WSCR)

For $WSCR$, we compare our approach with the baseline, Rahman, Peng, Opitz, Sakaguchi, and Human participants. The reported accuracies of Rahman, Peng, Opitz, and Sakaguchi on $WSCR$ are 73.05%, 76.76%, 63.0%, and 93.1% respectively. The human accuracy of 95.2% is taken from Sakaguchi et al.'s [43] report. Similar to the previous experiment, we compared the performance of four different methods of our approach with the others. The results are illustrated in Table 4.4.

Approach	Correct	Coverage	Problem Set	Relative Accuracy	Accuracy
Rahman	-	564	564	73.05	73.05
Peng	-	564	564	76.76	76.76
Opitz	-	564	564	63.00	63.00
Sakaguchi	-	564	564	93.10	93.10
Baseline	321	482	564	66.60	56.91
SW-L1	167	481	564	34.72	29.60
MW-L1	271	481	564	56.34	48.05
SW-L2	192	481	564	39.92	34.04
MW-L2	320	481	564	66.53	56.74
Human	-	564	564	95.2	95.2

Table 4.4: Performance Comparison: A. Rahman's Set (WSCR)

Our Approaches vs Others

From Table 4.4, we can see that all four methods of our approach are outperformed in terms of both relative and final accuracy. We also note that on this set, the baseline has much higher coverage compared to the Winograd Schema set (482 out of 564). However, another important observation is the lower accuracy of the baseline compared to the original Peng's approach. This shows that our modification of Peng's approach is not suitable for $WSCR$. We have analyzed the limitations of our approach and found two causes responsible for relatively lower performances.

- Multiple Associativity of Events: Our approach is suitable for sentences where the referents and pronouns are associated with a single event. In $WSCR$, we found a large number of problem instances where the entities are associated with multiple events. For example, one of them was: "Guns generally increase in value over time to gun collectors if unfired, because they need to be in mint condition to be extremely valuable". Here, "they" is associated with both "mint condition" and "extremely valuable" and "guns" is associated with both "increase in value" and "unfired". While it is reasonable to match the subject of "increase in value" with the subject of "extremely valuable", it is not reasonable to match the subject of "unfired" to the subject of "extremely valuable".

- Presence of non google-proof problems: $WSCR$ contains a large number of problem instances containing additional information that can be exploited by statistical approaches. For example, one of them was: "Bill Gates is similar to Steve Jobs, but he is the CEO of Microsoft". It is relatively easier for statistical approaches to match Bill Gates as the CEO of Microsoft. However, because this type of problems is guaranteed

to be removed from the Winograd Schema set, our approach contains no technique to exploit this kind of relationships between two proper nouns.

To judge the performance of our approach on a set with only google-proof examples, we manually remove all 79 non google-proof examples from $WSCR$ and create a smaller problem set $WSCRGP$. The results are illustrated in Table 4.5.

Approach	Correct	Coverage	Problem Set	Relative Accuracy	Accuracy
Baseline	286	440	485	65.00	58.97
SW-L1	151	423	485	35.70	31.13
MW-L1	253	423	485	59.81	52.16
SW-L2	171	423	485	40.43	35.26
MW-L2	297	423	485	70.21	61.24

Table 4.5: Performance Comparison: A.Rahman's Set Google-Proof (WSCRGP)

Table 4.5 shows that MW-L2 outperforms the baseline in terms of both relative and final accuracy in $WSCRGP$. This result proves that on a google-proof sample, accuracy of our approach increases.

It is important to note that among the two limitations we identified, the multiple associativity of events is still present in $WSCRGP$. Because of this limitation, we can conclude that even though our approach outperforms previous logic based approaches in the Winograd Schema set, it is still not reliable enough in general coreference resolution problem sets like $WSCR$. It remains as one of our future works to overcome this limitation.

Chapter 5

Discussion

In this thesis, we attempted to develop a system that is capable of using commonsense knowledge and performing logical reasoning to tackle one of the most challenging coreference resolution tasks: The Winograd Schema Challenge. For this task, we wanted to go beyond the lexical and syntactic analysis and use semantics and sentiment polarity for common verbs and adjectives. By analyzing the problem set as well as the previous works, we came to the conclusion that it is not possible to use a single logical model for the general Winograd Schema because of the diversity of the problem domain. Therefore, in this thesis, we only target the problems that involve causal reasoning. In this chapter, we present the contributions and limitations of our approach as well as the future works before concluding this thesis.

5.1 Contributions

This thesis provides a semantic approach to use polarities to solve coreference resolution problems. To the best of our knowledge, this is an initial step towards a new direction to understand the natural language from the viewpoint of human emotion.

We have addressed the background of the Winograd Schema challenge and both the statistical and logical approaches. Moreover, we provide a comparative analysis of the strengths and weaknesses of previous works and explain why they can not achieve human-like accuracy.

In this thesis, we have developed a model where polarity is expressed as a vector of different features (e.g., "ethics", "ability"). In order to create the most effective feature vector, we have analyzed Tuvery et al.'s [51] feature set and measure their relative importance as well as their inter-dependencies.

As a part of our approach to the Winograd Schema Challenge, we have created a knowledge base that stores the polarities of common verbs and adjectives. Unfortunately, we can not validate the accuracy of the knowledge base because it has a different structure compared to existing similar knowledge bases, and also it is too large to judge manually.

Our approach is an initial step to analyze and solve different coreference resolution problems using semantics and sentiment polarity. We compare its performance with the performances of both the recent statistical and logical methods on two benchmark problem sets $WSCL$ and $WSCR$. The experimental results show that this approach has the potential to disambiguate some problems where lexical and syntactic analysis are not enough.

5.2 Future Works

Our approach has some limitations that can lead to possible future extensions. Based on the experimental results, we have found some possible enhancements which can enable our approach to perform better for less structured problems.

We are still outperformed by the state of the art statistical approach. However, as pointed out by Sakaguchi et al. [43], this performance gap is most likely because of some of the inherent bias in the problem set. Testing our approach on a bias free dataset like "$WinoGrand_{debiased}$" [43] is one of the possible future works of this thesis.

We consider at most two independent features for polarity: "ability" and "ethics". It is clear from Tuvery's [51] classification that there are several other independent features that we refined in section 3.1. We also show in this thesis that using a larger number of independent feature helps the reasoning module. It can be a possible extension of our work to break down the "ability" feature to form a larger feature vector on the Winograd Schema problems.

Because of the lack of properly and reliably annotated data set, we can not measure the accuracy of our knowledge base. This has a negative effect on the coreference resolution problem for both $WSCL$ and $WSCR$ problem sets. With proper human annotated polarity values, it is possible to evaluate the correctness of the knowledge base.

Our assumption of each event being either verb dominant, trait dominant, reverse verb dominant, or idiom is valid for a limited number of cases. It is difficult to cover all possible cases with any logic based method. A possible remedy is to use Recurrent Neural Networks (RNNs). However, training an RNN requires a large amount of data. Therefore it is a possible future work to create a suitable training set and use supervised techniques like RNN and LSTM to determine polarities for arbitrary length text sequences.

Our approach is only applicable for causal reasoning, where one of the events acts as a cause to the other. However, the Winograd Schema set also contains problems that have no such relationship. It is possible to extend our approach to support problems that require different types of reasoning and thereby increasing the coverage.

Based on our experiment on $WSCR$ problem set, we found that in the presence of non google-proof examples, our approach is outperformed by the statistical approaches. However, after removing the non google-proof examples, the performance improves. It is possible to combine our semantic approach with the statistical approaches by converting

our rules into constrains and developing a loss function based on the constraints. A similar hybrid approach was used by Liu et al. [21]. It may work well for less structured problem sets than $WSCL$.

5.3 Conclusion

The Winograd Schema Challenge is currently one of the most reliable methods to judge the intelligence of an artificial agent. The challenge involves a short text involving two parties and a pronoun. The agent is asked to match the pronoun with the correct party. The problems are carefully designed so that there is no purely statistical approach to reliably solve it without semantic analysis and commonsense logic. However, at present, commonsense logic based methods are significantly outperformed by the statistical methods. In this thesis, we have presented a detailed analysis of the previous approaches to the Winograd Schema. The objective of this thesis is to find a commonsense based solution to a specific form of the Winograd Schema problem set that involves understanding the sentiment polarity of the important words of the text.

This thesis shows that the polarity of a word itself is a combination of many independent features, and each feature value can be logically determined from the polarities of surrounding words, which appears in the same context of the target word. Based on this observation, we have analyzed different features of polarity and their inter-dependency, and constructed an independent feature set. Although we are not able to implement the full feature set, we have shown in this thesis that using a higher number of independent features has a positive influence on coreference resolution accuracy.

As a part of our approach to the Winograd Schema, we have created a knowledge base to store the polarities of verbs and adjectives. Moreover, we have developed a simple rule based method to derive the polarities of events from their constituent words that allows us to determine the polarity of each event of the Winograd Schema from our knowledge base. From the events and their relationships to associated entities, we have developed a rule-based method to determine the correct candidate by following a set of inference steps.

We have compared the performance of our approach to the existing approaches on a subset of Winograd Schema set $WSCL$, where events are causally related. In addition to the Winograd Schema set, we have also applied it on a slightly different pronoun disambiguation problem set $WSCR$ to observe its effectiveness on a more diverse set of problems. The results show that our approach with multidimensional polarity outperforms previous logic based approaches but is outperformed by the state of the art statistical approach on both problem sets. However, after refining the problems of $WSCR$, the accuracy of our approach has increased significantly.

Bibliography

[1] Daniel Bailey, Amelia J Harrison, Yuliya Lierler, Vladimir Lifschitz, and Julian Michael. The Winograd Schema Challenge and reasoning about correlation. In *2015 AAAI Spring Symposium Series*, 2015.

[2] David Bender. Establishing a human baseline for the Winograd Schema Challenge. In *MAICS*, pages 39–45, 2015.

[3] Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical papers*, pages 2666–2677, 2016.

[4] Erik Cambria and Bebo White. Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2):48–57, 2014.

[5] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. *Proceedings of ACL-08*, pages 789–797, 2008.

[6] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461, 2017.

[7] Albert T Corbett and Frederick R Chang. Pronoun disambiguation: Accessing potential antecedents. *Memory & Cognition*, 11(3):283–294, 1983.

[8] Ernest Davis, Leora Morgenstern, and Charles L Ortiz. The first Winograd Schema Challenge at IJCAI-16. *AI Magazine*, 38(3):97–98, 2017.

[9] Ernest Davis, Leora Morgenstern, and Charles L Ortiz. The first Winograd Schema Challenge at IJCAI-16. *AI Magazine*, 38(3):97–98, 2017.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[11] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics (volume 2: Short papers)*, pages 49–54, 2014.

[12] Ali Emami, Noelia De La Cruz, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. A Knowledge Hunting Framework for Common Sense Reasoning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1958, 2018.

[13] Ali Emami, Paul Trichelair, Adam Trischler, Kaheer Suleman, Hannes Schulz, and Jackie Chi Kit Cheung. The KnowRef Coreference Corpus: Removing Gender and Number Cues for Difficult Pronominal Anaphora Resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3952–3961, 2019.

[14] Samuel D Gosling, Peter J Rentfrow, and William B Swann Jr. A very brief measure of the Big-Five personality domains. *Journal of Research in Personality*, 37(6):504–528, 2003.

[15] Pengcheng He, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. A Hybrid Neural Network Model for Commonsense Reasoning. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 13–21, 2019.

[16] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

[17] Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. A Surprisingly Robust Trick for the Winograd Schema Challenge. Association for Computational Linguistics, 2019.

[18] Himabindu Lakkaraju, Richard Socher, and Chris Manning. Aspect specific sentiment analysis using hierarchical deep learning. In *NIPS Workshop on deep learning and representation learning*, 2014.

[19] Hector Levesque, Ernest Davis, and Leora Morgenstern. The Winograd Schema Challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[20] Hugo Liu and Push Singh. ConceptNet: A practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, 2004.

[21] Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. Combing context and commonsense knowledge through neural networks for solving Winograd Schema problems. In *2017 AAAI Spring Symposium Series*, 2017.

[22] Edward Loper and Steven Bird. NLTK: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[23] Yukun Ma, Haiyun Peng, and Erik Cambria. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[24] Maryellen C MacDonald, Neal J Pearlmutter, and Mark S Seidenberg. The lexical nature of syntactic ambiguity resolution. *Psychological Review*, 101(4):676, 1994.

[25] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, 2017.

[26] Inc Merriam-Webster. *Webster's ninth new collegiate dictionary*. Merriam-Webster, 1983.

[27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[29] George A Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[30] Leora Morgenstern, Ernest Davis, and Charles L Ortiz. Planning, executing, and evaluating the Winograd Schema Challenge. *AI Magazine*, 37(1):50–54, 2016.

[31] Leora Morgenstern and Charles Ortiz. The Winograd Schema Challenge: evaluating progress in commonsense reasoning. In *Twenty-Seventh IAAI Conference*, 2015.

[32] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture*, pages 70–77. ACM, 2003.

[33] Thien Hai Nguyen and Kiyoaki Shirai. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2514, 2015.

[34] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. Using of Jaccard coefficient for keywords similarity. In *Proceedings of the International Multiconference of Engineers and Computer Scientists*, volume 1, pages 380–384, 2013.

[35] Juri Opitz and Anette Frank. Addressing the Winograd Schema Challenge as a sequence ranking task. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 41–52, 2018.

[36] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[37] Haoruo Peng, Daniel Khashabi, and Dan Roth. Solving hard coreference problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819, 2015.

[38] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, and Orphée De Clercq. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, 2016.

[39] Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: the Winograd Schema Challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789. Association for Computational Linguistics, 2012.

[40] Adam Richard-Bollans, Lucıa G Alvarez, and Anthony G Cohn. The role of pragmatics in solving the Winograd schema challenge. In *Proceedings of the Thirteenth International Symposium on Commonsense Reasoning (Commonsense 2017)*. CEUR Workshop Proceedings, 2018.

[41] Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. Gender Bias in Coreference Resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, 2018.

[42] Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1546–1556, 2016.

[43] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WINOGRANDE: An Adversarial Winograd Schema Challenge at Scale.

[44] Roger C Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive psychology*, 3(4):552–631, 1972.

[45] Peter Schüller. Tackling Winograd Schemas by formalizing relevance theory in knowledge graphs. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2014.

[46] Arpit Sharma, Nguyen H Vo, Somak Aditya, and Chitta Baral. Towards addressing the Winograd Schema Challenge: Building and using a semantic parser and a knowledge hunting module. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[47] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective LSTMs for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, 2016.

[48] Tun Thura Thet, Jin-Cheon Na, and Christopher SG Khoo. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, 36(6):823–848, 2010.

[49] Paul Trichelair, Ali Emami, Jackie Chi Kit Cheung, Adam Trischler, Kaheer Suleman, and Fernando Diaz. On the evaluation of common-sense reasoning in natural language understanding. *NeurIPS Workshop on Critiquing and Correcting Trends in Machine Learning*, 2018.

[50] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*, 2018.

[51] Franco Tuveri and Manuela Angioni. A linguistic approach to feature extraction based on a lexical database of the properties of adjectives and adverbs. In *GWC 2012 6th International Global Wordnet Conference*, page 365, 2012.

[52] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems.

[53] Michael White, Mark Steedman, Jason Baldridge, and Geert-Jan Kruijff. OpenCCG: The OpenNLP CCG Library. 2008.

[54] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. OpinionFinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, 2005.

[55] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[56] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.