A Generalized Knowledge Hunting Framework for the Winograd Schema Challenge

Ali Emami[1], Adam Trischler[2], Kaheer Suleman[2], and Jackie Chi Kit Cheung[1]

[1]School of Computer Science, McGill University
[2]Maluuba Research, Montreal, Quebec
ali.emami@mail.mcgill.ca
{*adam.trischler, kasulema*}*@microsoft.com*
jcheung@cs.mcgill.ca

Abstract

The Winograd Schema Challenge is a popular alternative Turing test, comprising a binary-choice coreference-resolution task that requires significant common-sense and world knowledge to solve. In this paper, we propose a novel framework that successfully resolves many Winograd questions while imposing minimal restrictions on their form and difficulty. Our method works by (i) generating queries from a parsed representation of a Winograd question, (ii) acquiring relevant knowledge using Information Retrieval, and (iii) reasoning on the gathered knowledge. Our approach improves the F1 performance by 0.16 over previous works, without task-specific supervised training.

1 Introduction

The Winograd Schema Challenge (WSC) has emerged as a popular alternative to the Turing test as a means to measure progress towards human-like artificial intelligence (Levesque et al., 2011). WSC problems are short passages containing a target pronoun that must be correctly resolved to one of two possible antecedents. They come in pairs which differ slightly and result in different correct resolutions. As an example:

(1) a. Jim yelled at Kevin because *he* was so upset. (Answer: Jim)
b. Jim comforted Kevin because *he* was so upset. (Answer: Kevin)

WSC problem pairs ("twins," using the terminology of Hirst (1988)) are carefully controlled such that heuristics involving syntactic salience, the number and gender of the antecedent, or other simple syntactic and semantic cues are ineffective. This distinguishes the task from the standard coreference resolution problem. Performant systems must make common-sense inferences; i.e., that someone who yells is likely to be upset, and that someone who is upset tends to be comforted. Additional examples are shown in Table 1.

WSC problems are simple for people to solve but difficult for automatic systems because common-sense reasoning encompasses many types of reasoning (causal, spatio-temporal, etc.) and requires a wide breadth of knowledge. There have been efforts to encode such knowledge directly, using logical formalisms (Bailey et al., 2015) or by using deep learning models (Liu et al., 2016a); however, these approaches have so far solved only restricted subsets of WSC questions with high precision, and show limited ability to generalize to new instances. Other work aims to develop a repository of common-sense knowledge (e.g., Cyc (Lenat, 1995), ConceptNet (Liu and Singh, 2004)) using semi-automatic methods. These knowledge bases are necessarily incomplete and further processing is required to retrieve the entries relevant to a given WSC context. Even given the appropriate entries, further reasoning operations must usually be performed as in Liu et al. (2016b); Huang and Luo (2017).

In this work we propose a three-stage knowledge hunting method for solving the WSC. We hypothesize that on-the-fly, large-scale processing of textual data can complement knowledge engineering efforts to automate common-sense reasoning. In this view, information that appears in natural text can act as implicit or explicit evidence for the truth of candidate WSC resolutions.

There are several challenges inherent to such an approach. First, WSC instances are explicitly designed to be robust to the type of statistical correlations that underpin modern distributional lexical semantics. In the example above, *yelled at* and *comforted* are both similar to *upset*, so it is difficult to distinguish the two cases by lexical similarity. Also, common sense involves background

Proceedings of NAACL-HLT 2018: Student Research Workshop, pages 25–31
New Orleans, Louisiana, June 2 - 4, 2018.

1 a)	The man couldn't lift his son because he was so weak. (Answer: the man)
1 b)	The man couldn't lift his son because he was so heavy. (Answer: son)
2 a)	The older students were bullying the younger ones, so we punished them. (Answer: the older students)
2 a)	The older students were bullying the younger ones, so we rescued them. (Answer: the younger ones)
3 a)	Sam tried to paint a picture of shepherds with sheep, but they ended up looking more like golfers. (Answer: shepherds)
3 b)	Sam tried to paint a picture of shepherds with sheep, but they ended up looking more like dogs. (Answer: sheep)

Table 1: Examples of Winograd Questions.

knowledge that is, by definition, shared by most readers. Common sense is thus assumed knowledge that is rarely stated explicitly in naturally occurring text. As such, even modern NLP corpora composed of billions of word tokens, like Gigaword (Graff and Cieri, 2003) and Google News (*http://news.google.com*), are unlikely to offer good coverage – or if they do, instances of specific knowledge are likely to be diffuse and rare ("long tail").

Information Retrieval (IR) techniques can sidestep some of these issues by using the *entire* indexed Internet as an input corpus. In particular, our method of **knowledge hunting** aims to retrieve scenarios that are similar to a given WSC question but where the ambiguities built into the question are absent. For example, to solve (1a), the following search result contains the relevant knowledge without the matching ambiguity:

(2) I got really upset with her and I started to yell at her because...

Here, the same entity *I* is the subject of both *upset* and *yell at*, which is strong evidence for resolving the original ambiguity. This information can be extracted from a syntactic parse of the passage using standard NLP tools.

Previous work on end-to-end knowledge-hunting mechanisms for the WSC includes a recent framework that compares query counts of evidence retrieved online for the competing antecedents (Sharma et al., 2015). That framework's coverage is restricted to a small subset of the Winograd instances based on knowledge constraints. In contrast, our approach covers a much larger subset of WSC passages and is impartial to knowledge constraints. Our framework adopts a novel representation schema that achieves significant coverage on Winograd instances, as well as an antecedent selection process that considers the evidence strength of the knowledge retrieved to make a more precise coreference decision.

Our method achieves a balanced F1 of 0.46 on the WSC, which significantly improves over the previous state-of-the-art of 0.3. We will also discuss the importance of F1 as a basis for comparing systems on the WSC, since it prevents overspecifying systems to perform well on certain WSC instances (boosting precision at the cost of recall).

2 Knowledge Hunting Framework

Our framework takes as input a Winograd sentence and processes it through three stages that culminate in the final coreference decision. First, it fits the sentence to a semantic representation schema and generates a set of queries that capture the predicates in the sentence's clauses. The query set is then sent to a search engine to retrieve text snippets that closely match the schema. Finally, returned snippets are resolved to their respective antecedents and the results are mapped to a best guess for the original Winograd question's resolution. We detail these stages below.

2.1 Semantic Representation Schema

The first step of our system is to perform a partial parse of each sentence into a shallow semantic representation; that is, a general skeleton of each of the important semantic components in the order that they appear.

In general, Winograd questions can be separated into a *context* clause, which introduces the two competing antecedents, and a *query* clause, which contains the target pronoun to be resolved. We use the following notation to define the components in our representation schema:

E_1, E_2	the candidate antecedents
$Pred_C$	the context predicate
$+$	discourse connective
P	the target pronoun
$Pred_Q$	the query predicate

E_1 and E_2 are noun phrases in the sentence. In the WSC, these two are specified and can be

identified without ambiguity. $Pred_C$ is the context predicate composed of the verb phrase relating both antecedents to some event. The context contains E_1, E_2, and the context predicate $Pred_C$. The context and the query clauses are often connected by a discourse connective $+$. The query contains the target pronoun, P, which is also specified unambiguously. In addition, preceding or succeeding P is the query predicate, $Pred_Q$, a verb phrase involving the target pronoun. Table 2 shows sentence pairs in terms of each of these components.

2.2 Query Generation

In query generation, we aim to generate queries to send to a search engine in order to extract text snippets that resemble the original Winograd sentence. Queries are of the form:

$+Term_C$ $+Term_Q$ $-$"Winograd"$-E_1$

We assume here that the search queries are composed of two fundamental components, $Term_C$ and $Term_Q$, which are strings that represent the events occurring in the first (context) and second (query) clause of the sentence, respectively. In addition, by excluding search results that may contain *Winograd* or E_1, we ensure that we do not retrieve some rewording of the original Winograd sentence itself.

The task then is to construct the two query sets, C and Q, whose elements are possible entries for $Term_C$ and $Term_Q$, respectively. We achieve this by identifying the root verbs along with any modifying adjective in the context and query clauses, using Stanford CoreNLP's dependency parse of the sentence. We then add the root verbs and adjectives into the sets C and Q along with their broader verb phrases (again identified directly using the dependency tree). These extracted queries serve as event information that will be used in the subsequent modules. Bean and Riloff (2004) also learn extraction patterns to support coreference, but unlike our method, their method relies on a static domain and constructs an explicit probabilistic model of the narrative chains learned.

Augmenting the query set with WordNet We use WordNet (Kilgarriff, 2000) to construct an augmented query set that contains synonyms for the verbs or adjectives involved in a representation. In particular, we include the synonyms listed for the top synset of the same part of speech as the extracted verb or adjective.

Manual query construction To understand the impact of the query generation step, we also manually extracted representations for all Winograd questions. We limited the size of these sets to five to prevent a blowing-up of search space during knowledge extraction.

In Table 3 we show examples of generated queries for C and Q using the various techniques.

2.3 Extracting Knowledge from Search Results

From the search results, we obtain a set of text snippets that sufficiently resemble the original Winograd sentence, as follows. First, $Term_C$ and $Term_Q$ are restricted to occur in the same snippet, but are allowed to occur in any order. We filter the resulting sentences further to ensure that they contain at least two entities that corefer to one another. These sentences may be structured as follows:

$$\begin{array}{lcl} E'_1\ Pred'_C\ E'_2 & + & E'_3\ Pred'_Q \\ E'_1\ Pred'_C\ E'_2 & + & Pred'_Q\ E'_3 \\ E'_1\ Pred'_C & + & E'_3\ Pred'_Q \\ E'_1\ Pred'_C & + & Pred'_Q\ E'_3 \end{array}$$

We call these *evidence sentences*. They exhibit a structure similar to the corresponding Winograd question, but with different entities and event order. In particular, $Pred'_C$ and $Pred'_Q$ (resulting from the queries $Term_C$ and $Term_Q$, resp.) should ideally be similar if not identical to $Pred_C$ and $Pred_Q$ from the original Winograd sentence. Note, however, that E'_1, E'_2, and E'_3 may not all have the same semantic type, potentially simplifying their coreference resolution and implying the correct resolution of their Winograd counterpart.

A sentence for which E'_3 refers to E'_1 is subsequently called an *evidence-agent*, and one for which E'_3 refers to E'_2 an *evidence-patient*. The exception to this rule is when an event occurs in the passive voice (e.g., *was called*), which reverses the conventional order of the agent and patient: where in active voice, the agent precedes the predicate, in passive voice, it succeeds it. Another exception is in the case of *causative alternation*, where a verb can be used both transitively and intransitively. The latter case can also reverse the conventional order of the agent and patient (e.g., *he opened the door* versus *the door opened*).

As an example of the previously mentioned

Pair	$Pred_C$	E_1	E_2	$Pred_Q$	P	Alternating Word (POS)
1	couldn't lift	the man	his son	was so heavy	he	weak/heavy (adjective)
2	were bullying	the older students	the younger ones	punished	them	punished/rescued (verb)
3	tried to paint	shepherds	sheep	ended up .. like	they	golfers/dogs (noun)

Table 2: Winograd sentence pairs from Table 1.

Sentence: The trophy doesn't fit into the brown suitcase because it is too large.

Query Generation Method	C	Q
Automatic	{"doesn't fit into", "brown", "fit" }	{"large", "is too large"}
Automatic, with synonyms	{"doesn't fit into", "brown", "accommodate", "fit", "suit" }	{"large", "big", "is too large" }
Manual	{"doesn't fit into", "fit into","doesn't fit" }	{"is too large", "too large"}

Table 3: Query generation techniques on an example Winograd sentences, where C and Q represent the sets of queries that capture the context and query clauses of the sentence, respectively.

coreference simplification, a valid evidence sentence is: *He tried to call **her** but **she** wasn't available.* Here, the sentence can be resolved simply on the basis of the gender of the antecedents; E'_3 – in this case, the pronoun *she* – refers to the patient, E'_2. Accordingly, the sentence is considered an evidence-patient.

2.4 Antecedent Selection

We collect and reason about the set of sentences acquired through knowledge extraction using a selection process that a) resolves E'_3 in each of these sentences to either E'_1 or E'_2 (rendering them either evidence-agent or evidence-patient), by direct use of CoreNLP's coreference resolution module; and b) uses both the count and individual features of the evidence sentences to resolve a given Winograd sentence. For example, the more similar evidence-**agents** there are for the sentence *Paul tried to call George on the phone, but he wasn't successful*, the more likely it is that the process would guess *Paul*, the **agent**, to be the correct referent of the target pronoun.

To map each sentence to either an evidence-agent or evidence-patient, we developed a rule-based algorithm that uses the syntactic parse of an input sentence. This algorithm outputs an evidence label along with a list of features.

The features indicate: which two entities corefer according to Stanford CoreNLP's resolver, and to which category of E'_1, E'_2, or E'_3 each belong; the token length of the sentence's search terms, $Term_C$ and $Term_Q$; the order of the sentence's search terms; whether the sentence is in active or passive voice; and whether or not the verb is causative alternating. Some of these features are straightforward to extract (like token length and order, and coreferring entities given by CoreNLP), while others require various heuristics. To map each coreferring entity in the snippet to E'_1, E'_2, or E'_3 (corresponding loosely to context subject, context object, and query entity, respectively), we consider their position relative to the predicates in the original Winograd question. That is, E'_1 precedes $Term_C$, E'_2 succeeds $Term_C$, and E'_3 may precede or succeed $Term_Q$ depending on the Winograd question. To determine the voice, we use a list of auxiliary verbs and verb phrases (e.g., *was*, *had been*, *is*, *are being*) that switch the voice from active to passive (e.g., "they are being bullied" vs "they bullied") whenever one of these precedes $Term_C$ or $Term_Q$ (if they are verbs). Similarly, to identify causative alternation, we use a list of causative alternating verbs (e.g., *break*, *open*, *shut*) to identify the phenomenon whenever $Term_C$ or $Term_Q$ is used intransitively.

These features determine the evidence label, evidence-agent (EA) or evidence-patient (EP), according to the following rules:

$$Label(e) = \begin{cases} \text{EA}, & \text{if } E'_3 \text{ refers to } E'_1, \text{ active (1)} \\ \text{EA}, & \text{if } E'_3 \text{ refers to } E'_2, \text{ passive (2)} \\ \text{EP}, & \text{if } E'_3 \text{ refers to } E'_2, \text{ active (3)} \\ \text{EP}, & \text{if } E'_3 \text{ refers to } E'_1, \text{ passive (4)} \\ \text{EP}, & \text{if } E'_1 \text{ refers to } E'_3, \text{ causative (5)} \end{cases}$$

The exceptions, (2), (4), and (5), can be illustrated with the following examples:

- *The weight couldn't be lifted by me, because I was so weak.* Here, because of the passive voice, E_2' plays the agent role, while syntactically being the object. Using rule (2), the sentence is correctly reversed to evidence-agent.

- *The weight couldn't be lifted by me, because it was so heavy.* For similar reasons, the sentence is correctly reversed to evidence-patient by rule (4).

- *The weight lifted. It was heavy.* This is reversed to evidence-patient, since 'lift' is a causative alternating verb by rule (5).

In addition to determining the evidence label, the features are also used in a heuristic that generates scores we call *evidence strengths* for each evidence sentence, as follows:

$$Strength(e) = LengthScore(e) + OrderScore(e)$$

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

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

The final stage of our framework runs the above processes on all snippets retrieved for a Winograd sentence. The sum of strengths for the evidence-agents are compared to that of the evidence-patients to make a resolution decision.

3 Experiments

We tested three versions of our framework (varying in the method of query generation: automatic *vs.* automatic with synonyms *vs.* manual) on the original 273 Winograd sentences (135 pairs and one triple). We compared these systems with previous work on the basis of Precision (P), Recall (R), and F1, where precision is the fraction of correctly answered instances among answered instances, recall is the fraction of correctly answered instances among all instances, and

$$F1 = 2 * P * R/(P + R).$$

We used Stanford CoreNLP's coreference resolver (Raghunathan et al., 2010) during query generation to identify the predicates from the syntactic parse, as well as during antecedent selection to retrieve the coreference chain of a candidate evidence sentence. Python's Selenium package was used for web-scraping and Bing-USA and Google (top two pages per result) were the search engines (we unioned all results). The search results comprise a list of document snippets that contain the queries (for example, "yelled at" and "upset"). We then extract the sentence/s within each snippet that contain the query terms (with the added restriction that the terms should be within 70 characters of each other to ensure relevance). For example, for the queries "yelled at" and "upset", one snippet is: "Once the football players left the car, she testified that she yelled at the girl because she was upset with her actions from the night before."

In the next section we compare the performance of our framework with the most recent automatic system that tackles the original WSC (Sharma et al., 2015) (S2015). In addition to P/R/F1, we also compare systems' evidence coverage, by which we mean the number of Winograd questions for which evidence sentences are retrieved by the search engine. This should not be conflated with the *schemal* coverage of our system, by which we mean the number of Winograd questions that syntactically obey Class A (85% of the Winograd questions). Our system is designed specifically to resolve these Class A questions. We nevertheless test on the remaining 15% in our experiments.

Although other systems for the WSC exist outside of S2015, their results are not directly comparable to ours for one or more of the following reasons: a) they are directed towards solving the larger, easier dataset; b) they are not entirely automatic; or c) they are designed for a much smaller, author-selected subset of the WSC. We elaborate on this point in Section 5.

4 Results

Table 4 shows the precision, recall, and F1 of our framework's variants, automatically generated queries (AGQ), automatically generated queries with synonyms (AGQS), and manually generated queries (MGQ), and compares these to the systems of Sharma et al. (2015) (S2015) and Liu et al. (2016b) (L2016). The system developed by Liu et al. (2016b) uses elements extracted manually from the problem instances, so is most closely comparable to our MGQ method. Our best automated framework, AGQS, outperforms S2015 by 0.16 F1, achieving much higher recall (0.39 vs

	# Correct	P	R	F1
AGQ	73	0.53	0.27	0.36
AGQS	106	0.56	**0.39**	**0.46**
S2015	49	**0.92**	0.18	0.30
Systems with manual information:				
L2016	43	0.61	0.15	0.25
MGQ	118	0.60	0.43	0.50

Table 4: Coverage and performance on the Winograd Challenge (273 sentences). The best system on each measure is shown in bold.

0.18). Our results show that the framework using manually generated queries (MGQ) performs best, with an F1 of 0.50. We emphasize here that the promise of our approach lies mainly in its generality, shown in its improved coverage of the original problem set: it produces an answer for 70% of the instances. This coverage surpasses previous methods, which only admit specific instance types, by nearly 50%.

The random baseline on this task achieves a P/R/F1 of .5. We could artificially raise the F1 performance of all systems to be above .5 by randomly guessing an answer in cases where the system makes no decision. We chose not to do this so that automatic systems are compared transparently based on when they decide to make a prediction.

5 Related work

All the IR approaches to date that have tackled the Winograd Schema Problem have done so in one of two ways. On the one hand, some systems have been developed exclusively for Rahman and Ng's expanded Winograd corpus, achieving performance much higher than baseline. Bean and Riloff (2004) learn domain-specific narrative chains by bootstrapping from a small set of coreferent noun pairs. Conversely, other systems are directed towards the original, more difficult Winograd questions. These systems demonstrate higher-than-baseline performance but only on a small, author-selected sub-set, where the selection is based often on some knowledge-type constraints.

Systems directed exclusively towards the expanded Winograd corpus include Rahman and Ng's system itself (Rahman and Ng, 2012), reporting 73% accuracy on Winograd-like sentences, and Peng et al.'s system that improves accuracy to 76% (Peng et al., 2015). Another system uses sentence alignment of web query snippets to resolve the Winograd-like instances, reporting 70% accuracy on a small subset of the test sentences in the expanded corpus (Kruengkrai et al., 2014). Unfortunately, the passages in the original WSC confound these systems by ensuring that the antecedents themselves do not reveal the coreference answer. Many sentences in the expanded corpus can be resolved using similarity/association between candidate antecedents and the query predicate. One such sentence is "Lions eat zebras because they are predators." Many of the above systems simply query "Lions are predators" versus "zebras are predators" to make a decision.

This kind of exploitation is often the top contributor to such systems' overall accuracy (Rahman and Ng, 2012), but fails to hold for the majority (if not all of) the original Winograd questions. In these questions one vital property is enforced: that the question should not be "Google-able." Our work seeks to alleviate this issue by generating search queries that are based exclusively on the predicates of the Winograd sentence, and not the antecedents, as well as considering the strength of the evidence sentences.

The systems directed towars the Original Winograd questions include Schüller (2014), who use principles from relevance theory to show correct disambiguation of 4 of the Winograd instances; Sharma et al. (2015)'s knowledge-hunting module aimed at a subset of 71 instances that exhibit *causal* relationships; Liu et al. (2016a)'s neural association model, aimed at a similar causal subset of 70 Winograd instances, and for which events were extracted manually; and finally, a recent system by Huang and Luo (2017) directed at 49 selected Winograd questions. While these approaches demonstrate that difficult coreference problems can be resolved when they adhere to certain knowledge or structural constraints, we believe that such systems will fail to generalize to the majority of other coreference problems. This important factor often goes unnoticed in the literature when systems are compared only in terms of precision; accordingly, we propose and utilize F1-driven comparison that does not enable boosting precision at the cost of recall.

6 Conclusion

We developed a knowledge-hunting framework to tackle the Winograd Schema Challenge. Our

system involves a novel semantic representation schema and an antecedent selection process acting on web-search results. We evaluated the performance of our framework on the original problem set, demonstrating performance competitive with the state-of-the-art. Through analysis, we determined our query generation module to be a critical component of the framework.

Our query generation and antecedent selection processes could likely be enhanced by various Machine Learning approaches. This would require developing datasets that involve schema identification, query extraction, and knowledge acquisition for the purpose of training. As future work, we consider using the extensive set of sentences extracted by our knowledge hunting framework in order to develop a large-scale, Winograd-like corpus. In addition, we are currently working to develop deep neural network models that perform both knowledge acquisition and antecedent selection procedures in an end-to-end fashion.

References

Dan Bailey, Amelia Harrison, Yuliya Lierler, Vladimir Lifschitz, and Julian Michael. 2015. The winograd schema challenge and reasoning about correlation. In *In Working Notes of the Symposium on Logical Formalizations of Commonsense Reasoning*.

David Bean and Ellen Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

David Graff and C Cieri. 2003. English gigaword corpus. *Linguistic Data Consortium* .

Graeme Hirst. 1988. Semantic interpretation and ambiguity. *Artificial intelligence* 34(2):131–177.

Wenguan Huang and Xudong Luo. 2017. Commonsense reasoning in a deeper way: By discovering relations between predicates. In *ICAART (2)*. pages 407–414.

Adam Kilgarriff. 2000. Wordnet: An electronic lexical database.

Canasai Kruengkrai, Naoya Inoue, Jun Sugiura, and Kentaro Inui. 2014. An example-based approach to difficult pronoun resolution. In *PACLIC*. pages 358–367.

Douglas B Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*. volume 46, page 47.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal* 22(4):211–226.

Quan Liu, Hui Jiang, Andrew Evdokimov, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. 2016a. Probabilistic reasoning via deep learning: Neural association models. *arXiv preprint arXiv:1603.07704* .

Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. 2016b. Combing context and commonsense knowledge through neural networks for solving winograd schema problems. *arXiv preprint arXiv:1611.04146* .

Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving hard coreference problems. *Urbana* 51:61801.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 492–501.

Altaf Rahman and Vincent Ng. 2012. 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*. Association for Computational Linguistics, pages 777–789.

Peter Schüller. 2014. Tackling winograd schemas by formalizing relevance theory in knowledge graphs. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Arpit Sharma, Nguyen Ha Vo, Somak Aditya, and Chitta Baral. 2015. Towards addressing the winograd schema challenge-building and using a semantic parser and a knowledge hunting module. In *IJCAI*. pages 1319–1325.