Addressing the Winograd Schema Challenge as a Sequence Ranking Task

Juri Opitz and **Anette Frank**
Research Training Group AIPHES,
Leibniz ScienceCampus "Empirical Linguistics and Computational Language Modeling"
Department for Computational Linguistics
69120 Heidelberg
{opitz,frank}@cl.uni-heidelberg.de

Abstract

The Winograd Schema Challenge targets pronominal anaphora resolution problems which require the application of cognitive inference in combination with world knowledge. These problems are easy to solve for humans but most difficult to solve for machines. Computational models that previously addressed this task rely on syntactic preprocessing and incorporation of external knowledge by manually crafted features. We address the Winograd Schema Challenge from a new perspective as a *sequence ranking task*, and design a Siamese neural sequence ranking model which performs significantly better than a random baseline, even when *solely* trained on sequences of words. We evaluate against a baseline and a state-of-the-art system on two data sets and show that anonymization of noun phrase candidates strongly helps our model to generalize.

1 Introduction

The Winograd Schema Challenge (WSC) targets difficult pronoun resolution problems which are easy to resolve for humans, but represent a great challenge for AI systems because they require the application of cognitive inferencing in combination with world knowledge (Levesque et al., 2012; Levesque, 2014). It has been argued that a computer that is able to solve WS problems with human-like accuracy must be able to perform "human-like" reasoning and that the WSC can be seen as an alternative to the Turing test. Consider the following Winograd Schema (WS):

Example 1.1 ***The city councilmen*** *refused* ***the demonstrators*** *a permit because* ***they*** *feared violence.*

Both *city councilmen* and *demonstrators* agree in number and gender and even in semantic type, as both mentions refer to groups of humans (with political interests). While we could imagine a city with councilmen who approve violence and hence forbid a demonstration by peaceful protesters, this reading may appear nonsensical to most readers. Most humans will straightforwardly resolve the pronoun *they* to corefer with *the city councilmen*. Now consider the outcome of replacing a single word – the predicate *feared* – with the semantically related predicate *advocated*, yielding its *twin* sentence:

Example 1.2 ***The city councilmen*** *refused* ***the demonstrators*** *a permit because* ***they*** *advocated violence.*

With this change, the resolution is reversed: now *they* refers to *the demonstrators*. Humans may reason that city council men are naturally concerned with the well-being of their city and thus *they* are not in favor of a demonstration by protesters who advocate violence. Winograd problems as displayed in Examples 1.1 and 1.2 occur very rarely in natural language texts and cannot be properly resolved by traditional coreference resolution (CR) systems. The primary reason is that standard CR systems heavily rely on features such as gender or number agreement or mention-distance information. However, such features do not give away any knowledge that would be useful for resolving WS problems. Given a random baseline of 0.5 accuracy, the Stanford resolver (Lee et al.,), winner of the CoNLL 2011 Shared Task (Pradhan et al., 2011), achieves a sobering accuracy of 0.53 when facing Winograd Schema problems (Rahman and Ng, 2012). Lee et al. (2017) describe a state-of-the-art neural system for general neural coreference resolution and observe that, while trained on much more data than is available in

Proceedings of the First International Workshop on Language Cognition and Computational Models, pages 41–52
Santa Fe, New Mexico, United States, August 20, 2018.
https://doi.org/10.18653/v1/P17

the WSC, their system shows little advance in the uphill battle of resolving hard pronoun coreference problems that require world knowledge.

As our main contribution we are proposing a novel and very general take on the WSC task that we formulate as a *sequence ranking task* in a *neural Siamese sequence ranking model*. Moreover, we design *features derived from manually designed knowledge bases* and show how they can be integrated in this model. We investigate *anonymization of noun phrase candidates* that significantly enhances the generalization capacity of the model. We evaluate against baselines and a state-of-the-art (SOTA) system with special focus on the impact of different features and *propose connotation frames* as a novel feature for the WSC task. All Siamese model variants, even those trained on word sequences only, show significant improvements over the baseline on our main testing set. Our best performing model achieves 0.63 accuracy.

2 WSC Datasets and Related Work

Strict Data is Scarce: WSCL. Starting with the work by Levesque et al. (2012), a collection of (currently) 282 *strict* WS problems is maintained online[1], which will henceforth be referred to as WSCL. We make a distinction between *strict* and *relaxed* Winograd Schemata. Relaxed Winograd Schemata are problems which can be solved by computing simple corpus statistics. E.g., *The chimpanzee couldn't use Linux because it is an animal* is of the relaxed type because a simple google query returns significantly more results for *chimpanzee is an animal* than *Linux is an animal* (19,700 vs. 3 hits). Such relaxed, easy-to-solve examples are not contained in the WSCL data set, but do occur in the WSCR data set, described below. The problems in WSCL have an average length of 18 tokens. Some problems may consist of more than one sentence and require understanding across sentence boundaries.[2]

Relaxed Data: WSCR. The main dataset used in this work[3], which we refer to as WSCR, was published by Rahman and Ng (2012). The data was created by 30 undergraduate students. It comprises 943 twin sentences and comes already divided into training (70%) and test set (30%). As opposed to the WSCL data, WSCR comprises both strict and relaxed Winograd schemata. We found that it also contains sentences with no straightforward resolution, as in Ex. 2.1 and 2.2 (with gold antecedents underlined):

Example 2.1 ***Bob*** *likes to play with* ***Jimbo*** *because* ***he*** *loves playing.*

Example 2.2 ***The bus driver*** *yelled at* ***the kid*** *after* ***she*** *drove her vehicle.*

When we presented these problems to a class of students, close to half of them voted for the other reading in Example 2.1 (more than half in Example 2.2). This is reasonable, since the alternative reading (Jimbo loves playing) can be inferred from the fact that generally people like to play with someone who likes to play – rather than with someone who does not like to play. The alternative reading of Example 2.2 could be even more likely, since it makes perfect sense that when a kid tries to drive the bus driver's vehicle, the bus driver will get angry and might yell at the kid. When inspecting the data, we found that while notably having lower quality than WSCL, most sentences have a clearly preferred reading, which coheres with the gold annotation. The problems in WSCR seem less diverse as all consist of exactly one sentence and in every sentence we find at least one discourse connector or a comma connecting a main clause with the antecedent candidates to a sub-clause that contains the pronoun.

Feature- and Example-based Ranking. Together with the WSCR data set, Rahman and Ng (2012) also publicized the description of a linear ranking system that achieves 73% accuracy on the published data. The system relies on 8 features, which it uses to fit a SVM ranking model. Contrary to our work, all features depend on syntactic dependency annotation. While incorporating complex external knowledge resources such as FrameNet (Baker et al., 1998) or narrative chains (Chambers and Jurafsky, 2008), the most helpful feature turned out to be simple Google-queries, it significantly outperformed the random baseline with a considerable margin of 6% to the next best single feature. Kruengkrai et al. (2014) attempted to replicate parts of the system, selecting five features. Some of them were implemented

[1] `https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.xml`

[2] E.g. *It was a summer afternoon, and* ***the dog*** *was sitting in the middle of the lawn. After a while, it got up and moved to* ***a spot under the tree****, because* ***it*** *was hot.*

[3] url: `http://www.hlt.utdallas.edu/~vince/papers/emnlp12.html`

differently, e.g. instead of querying Google directly, the Google n-gram dataset (Brants and Franz, 2006) was used. The authors present a system that extracts representative examples from the web. Both systems were tested on a subset of the WSCR test set (for the problems where web examples were found). The reimplemented system yielded 0.56 accuracy while their own approach yielded 0.69 accuracy.

Integer Linear Program (ILP). Peng et al. (2015b) use an ILP (Schrijver, 1986) inference approach with a novel way of knowledge representation. Their system yields 0.76 accuracy on WSCR, which is the current state-of-the-art result on this data. In their approach "Predicate Schemas" are instantiated and scored using knowledge acquired from external knowledge bases compiled into constraints for a decision. Consider 'The bee landed on the flower because **it** {was hungry, had pollen}', where the gold resolution is that (i) *the bee was hungry* and (ii) *the flower had pollen*. A simple predicate schema for this problem is instantiated as $hungry(bee)$ vs. $hungry(flower)$ and $has\ pollen(flower)$ vs. $has\ pollen(bee)$. Scores for the instantiated predicates are then gathered from external knowledge sources such as Google[4].

Other Work on Difficult Coreference Resolution. Sharma et al. (2015) build a semantic parser and Schüller (2014) use syntactic dependency annotation and knowledge base linking in order to solve WSC problems. Both works use the Answer Set Programming language (ASP, cf. (Baral, 2003; Gelfond and Lifschitz, 1988)) on the generated abstract representations for reasoning about the correct antecedent. Sharma et al. (2015) for evaluation considers only causal attributive and direct causal events and Schüller (2014) performs experiments with only 4 twin problems for demonstration purposes.

We conclude that (i) all examined prior work focuses on either a specific subset of Winograd problems or/and is tested on only one specific data set, WSCL or WSCR but never both. Also (ii) we are the first to present an end-to-end WSC system which, contrary to all prior methods, does not rely on sophisticated preprocessing or linguistic annotation. (iii) We avoid heavy reliance on Google searches, which we argue the approaches of both Rahman and Ng (2012) and Peng et al. (2015b) suffer from. This is mainly due to two reasons: 1., Google has restricted automatic access to their search engine, making it difficult to solve more than a handful of pronoun resolution problems in short time without payment and, even more importantly 2., reproduction of results is impossible due to the nature of Google's search-algorithm as a black box – one cannot ensure to retrieve the exact same or even similar query results as previous authors. Our work, by contrast, does not rely on non-reproducible features and will be the first to present an end-to-end neural approach for addressing the WSC.

3 Framing the WSC as a Sequence Ranking Task

We propose a new view on Winograd problems by translating the problem to a sequence ranking or classification task that discriminates a preferred or plausible sentence reading from a very similar but dispreferred or implausible reading. The preferred reading emerges when we replace the pronoun with its coreferent gold antecedent noun phrase and the dispreferred reading emerges when we instead use the wrong antecedent as the replacement. For example, given the WS problem *Joe paid the detective after he received the final report on the case.*, we can derive the preferred reading:

Example 3.1 ***Joe*** *paid* ***the detective*** *after* ***Joe*** *received the final report on the case.*

and the clearly less preferred reading

Example 3.2 ***Joe*** *paid* ***the detective*** *after* ***the detective*** *received the final report on the case.*

Most humans easily come to understand that Example 3.1 is in line with common sense (preferred), while the second Example 3.2 seems somewhat bogus and less in line with common sense (dispreferred). Inserting the correct (incorrect) antecedent noun phrase in place of the pronoun converts a Winograd problem with alternative but clear pronoun resolutions into preferred and dispreferred readings.

Formal Description. Let a Winograd problem be defined as a tuple $(s, p, c^+, c^-) \in W$, where s is a sequence of tokens, p is the given anaphoric pronominal token and c^+ represents the correct and c^- the incorrect noun phrase antecedent. We design a function $f : W \rightarrow W'$, returning a tuple $(r^+, r^-) \in W'$, containing two sequences of tokens, where r^+ is the preferred reading of s and r^- is the dispreferred reading of s which are the result of replacing the anaphoric pronoun p in s with c^+ or c^-.[5]

[4]Note that *plants* and *bees* are both very likely to have pollen, the predicate schema may be prone to errors in this case.

[5]When the pronouns are possessive (his, her, their), we replace p in s with the genitive form of c^+ or c^-.

Discussion. We derive sentences without pronouns from sentences with pronouns by inserting the aforementioned corresponding noun. A motivation for this process is the assumption that pronouns 'stand for', 'replace' or are 'substitutes' for previously mentioned or understood noun phrases. Framing the problem as a sequence preference ranking task has two major advantages. First, by replacing the anaphor with one of the possible antecedents, we contextualize each of these candidates to the local context of the anaphor. This contextualization can be exploited in a neural end-to-end system that constructs a full sentence representation, including the (resolved) pronoun. Second, with the two alternative readings being constructed, we can define a model that determines which of the two readings is preferred, or can be considered more plausible. That is, we frame the task as a preference ranking task, as opposed to a categorical binary classification task. In sum, we argue that framing/formulating the task of Winograd sentence/problem resolution as a task of *comparing the plausibility of alternative readings* provides an appealing alternative to prior task formulations: It permits the application of hypothetically any type of sentence representation model to be applied out-of-the-box.

Note however that by no means we want to postulate that humans understand and resolve Winograd problems by internally comparing a pair of complete sentence representations with alternatively resolved pronouns. But what perhaps is also clear is that humans do not dependency parse the full sentence and then access knowledge bases weighting manually crafted mention features as commonly done in the WSC task (Rahman and Ng, 2012; Sharma et al., 2015).

4 Neural Sequence Models for the WSC

Having converted each WS problem into two highly similar yet different readings allows us to define a neural end-to-end model in at least two different ways. In a naïve formulation (**Naïve Model**), we can simply force a model to predict whether a specific reading is plausible or implausible (binary classification). Alternatively, we can also exploit the fact that the two readings – produced by replacing the anaphor with a candidate antecedent (see above) – are highly similarand frame the task as a sequence ranking problem and design a *relational* model that constructs two internal representations that are compared and ranked. We call this the **Siamese Model**.

Naïve Model. We encode a sequence of tokens with an embedding layer and a two-layered Bi-LSTM (Hochreiter and Schmidhuber, 1997) and use a logistic regression layer on top to predict whether the sentence – representing one or the other of the two possible readings – is accepted or not. For training, from each pair of readings indexed by $i = 1, ..., N$ we extract two training examples, where the preferred reading r_i^+ is assigned class 1, and the dispreferred reading r_i^- is assigned class 0. This model can be optimized by minimizing a standard binary cross-entropy loss. A disadvantage of this model is that the classifier is not explicitly optimized towards the goal of discriminating competing readings since during training accepted and inaccepted readings are isolated from each other.

Siamese Model. Similar to the Naïve Model, we encode a sequence of tokens with an embedding layer followed by two-layered Bi-LSTM and use a single SELU (Klambauer et al., 2017) unit on top that predicts a plausibility score $h_\theta(r)$ for a reading r, where θ are the parameters of the model. The model is mirrored and uses shared weights to process two different representations at the same time, one for each reading (Fig. 1). We compute two plausibility scores over a pair of readings for every training example, where the aim is to maximize the difference between the scores for the plausible sequences and the implausible ones. At inference, the resolution with highest plausibility score is chosen. We avoid decomposing a pair of readings into two independent training and testing examples as done in the naïve model and by feeding the model both sequences at the same time we directly optimize the model to assign the preferred reading a higher plausibility score compared to the dispreferred reading. This is reflected in the (totally differentiable) margin ranking loss, which we define as

$$\frac{1}{N}\sum_{i=1}^{N}\Big[1-\sigma\big(h_\theta(r_i^+)-h_\theta(r_i^-)\big)\Big], \tag{1}$$

where σ is the logistic function. The general architecture is outlined in Fig. 1 and lends itself naturally to the incorporation of at least two different types of additional input features.

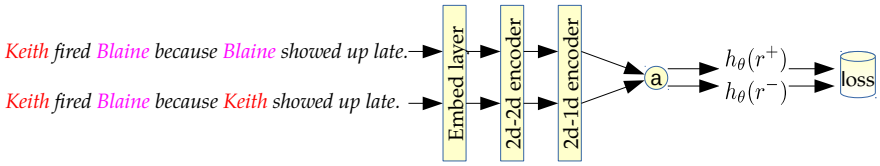

Figure 1: General Siamese architecture for comparing WSC readings. Embed layer is a function converting a sequence of tokens to a sequence of real valued vectors (we use a lookup table containing pretrained GloVe embeddings). 2d-2d encoder means any function that converts a sequence of vectors into another sequence of vectors (we use a Bi-LSTM returning state vectors). 2d-1d means any encoder converting a sequence of vectors into a single vector (we use a Bi-LSTM and concatenate the end states of each sequential read). 'a' can represent any activation neuron (we use a SELU unit).

Siamese Multi-Input Model. Our general architecture is displayed in Figure 1. The architecture naturally lends itself for the incorporation of many additional features, which have the potential to provide pointed world knowledge for the model that it cannot derive from the scarce training data. In the basic model (Figure 1), we can inject two additional types of features: real valued vectors and real valued matrices. Consider that the word embedding sequence for a Winograd example is of length l which is again projected by a Bi-LSTM (2d-2d encoder in Figure 1) onto a state matrix of dimension $l \times n$ and consider the case of one additional matrix type feature: after the matrix has been shaped to the same dimensionality $l \times n$ we can use concatenation, element-wise addition and element-wise multiplication to merge the additional feature representation with the sentence representation into a representation of dimension $l \times 4n$ before it is fed into the next layer. As additional matrix-type features we experiment with dependency edge sequences and information about the connotation of the arguments induced by their predicate as stated in the resource *Connotation Frames* (Rashkin et al., 2015; Rashkin et al., 2016). The features and motivation for usage are more extensively discussed in the next paragraphs.

We can also incorporate features which come as real valued vectors: we use an averaged semantic embedding of the tokens of the candidate noun phrase (described more closely in the next paragraph) to provide useful information for cases where the candidate noun phrase is not a generic person name but carries meaning. The vector can be injected into the model between the 2d-1d encoder and the output activation computation. A FF-layer is used to shape the vector so that it matches the output dimension h of the 2d-1d encoder enabling us to perform element-wise addition, element-wise multiplication and concatenation resulting in a high level sentence representation of dimension $4h$.

Anonymization of Candidate NPs. The fact that training data is really small motivates us to propose anonymization of noun phrase candidates as a simple means for discouraging the model to memorize the noun phrase candidates, forcing it to focus on the complex but general interactions between arguments and predicates. Consider the following pair (correct antecedents underlined):

Example 4.1 ***Mary*** *thanked* ***Susan*** *for all the help* ***she*** *had received.*

Example 4.2 ***Mary*** *thanked* ***Susan*** *for all the help* ***she*** *had given.*

Memorizing the candidates would be fatal for any model, since the resolution is not determined by the candidate noun phrases alone (both are generic names of the same gender), but rather from the interaction between predicates and arguments. We want the model to focus on deeper information from the meaning of the sentences that is general and relevant to support the correct resolution of the pronoun.

Candidate NP-level Feature. While many WS problems can be easily solved by humans in anonymized form, there are cases for which information about the candidate noun phrases is necessary, or even mandatory, especially for the relaxed Winograd problems in the WSCR-data. Consider

Example 4.3 *He hates* ***Cuba*** *and likes* ***Japan*** *because* ***it*** *is a communist country.*

This example is not strict because it is rather easy to resolve for machines by simply computing similarity measures between the candidate noun phrases and the predicate *communist country*.[6] ConceptNet (Speer and Havasi, 2012) and the available semantic embeddings trained on this resource (Speer and Lowry-Duda, 2017) i.a. contain information from WordNet (Miller, 1995) and may give the model the

[6]More precisely, the predicate *communist country* restricts the arguments unambiguously to the correct phrase.

```
S: (n) Cuba, Republic of Cuba (a communist state in the Caribbean on the island of
       Cuba)
S: (n) Cuba (the largest island in the West Indies)
```

Figure 2: WordNet gloss for the noun *Cuba*. It contains information about the political stance of the government.

	The	ball	hit	the	window	and	Bill	fixed	the	it.
p(wx)		-0.33			-0.20		0.33			-0.03
p(rx)		0.06			-0.73		0.13			0.40
e(x)		-0.20			0.33		0.60			0.73

Figure 3: An example of how we apply Connotation Frames for *hit* and *fix*. The numerical value $\in [-1, 1]$ ranges from positive (+1) to negative (-1). p(wx) and p(rx) represent the perspective of the writer (w) or reader (r) towards the object or subject of the verb x. e(x) stands for the effect on the subject or object. The frames contain 4 more perspectives which are omitted in the Figure.

information that *Cuba* is a communist country (see Figure 2). The information from the gloss that Cuba is the largest island in the West Indies is not necessary, but one could easily make up a WS problem for which it is necessary as in *He likes Cuba and hates Japan because it is located in the Caribbean Sea.*. This is the only feature acting on a candidate noun phrase level and it is computed by averaging the semantic embeddings of the corresponding tokenized candidate noun phrases.

Dependency Edges. For the resolution of many WS problems, feeding explicit syntactic information may be useful and help the model in learning useful information about predicates and their interactions. Consider Examples 4.1 and 4.2, where the predicate of the pronoun is *gives(x,help)* and the predicate in which both possible antecedents participate in is *thanks(Mary,Susan)*. It is very useful to know that Mary is the subject of *thanks* and Susan the object. When provided with such information the model may learn the abstract pattern that the subject argument of *gives* is more likely to be the object of *thank(x,y)* than its subject, while the subject argument of *receive* is more likely to be the subject argument of *thank(x,y)*.

Connotation Frames. *Connotation Frames* is a resource[7] that contains frames of verbs that indicate how the arguments of the verb are affected by the predicate meaning (Rashkin et al., 2015; Rashkin et al., 2016). The frames represent this information by presenting numerical values for seven types of connotations concerning different components of the frame. For example, the value of the object of the frame *resolve(s,o)* is negatively connotated. This reflects that what needs be to resolved is usually considered a problem, and a problem is most likely an issue which is perceived negatively. Consider

Example 4.4 ***The ball*** *hit* ***the window*** *and Bill fixed* ***it***.

For application we retrieve the frames for *hit* and *fix* and apply them to the arguments of the respective verbs in a sentence, resulting in a matrix with columns of dimension seven. The result is displayed in Figure 3 (where only 3 dimensions are displayed). For words or arguments of predicates not covered by the resource, we use a zero-vector.

5 Experiments

Data. Unlike most other research on WSC, we test our models on both data sets discussed above – WSCL, the smaller data set of higher quality (282 examples) with strict and mostly unambiguous WSC cases, which we exclusively use for testing and WSCR, which comes in a predefined split of 1322 training and 564 testing problems, but which is of slightly reduced quality for the reasons discussed in Section 3. Note that, as in previous work, we do not exploit the fact that each Winograd problem has a twin.

Baselines. Given that the WS problems in WSCL and WSCR come in pairs with alternative resolutions to first vs. second antecedent candidate, we apply a random process as the baseline with 0.5 probability of achieving the correct guess. Since the problem can be seen as a binary classification task, we calculate binomial tests to assess the probability of the zero-hypothesis that a random process achieves the same

[7] Available at `https://homes.cs.washington.edu/~hrashkin/connframe.html`.

amount or more correct predictions than the evaluated system. We also downloaded the state of the art system of Peng et al. (2015a), which the authors made publicly available[8]. However, it is important to note that the publicized system had been retrained on both training and testing data of WSCR[9], making it difficult to re-evaluate it under the original experimental conditions. When evaluating the system with anonymized candidates, we only select cases where the integrated mention detection was able to detect both (and only both) candidates and linked the pronoun to one of those. All other cases we have to treat as unresolved. The downloaded system yields an accuracy of 0.99 (397 correct, 3 incorrect, 164 unresolved) on WSCR. In our evaluation Table we present the result from their paper (Table 1: *SOTA*) As an additional baseline we use as a representation of the input sentences the representations predicted by a trained *sentence embedding model*, here *InferSent* (Conneau et al., 2017). InferSent has been been trained on large-scale natural language inference tasks (Bowman et al., 2015) and therefore may have internalized valuable information about whether sentence readings are coherent or rather nonsensical. We infer 4096-dimensional sentence vectors with the trained model provided by the authors[10] and fit a linear ranker SVM, using randomly sampled development data to find a suitable regularization parameter.

Experimental Setup and Evaluation. We evaluate our models in two testing scenarios: (i) *Train:*WSCR+*Test:*WSCL*:* In this setup we train the model on the full WSCR data and test on the unseen WSCL data, to test the generalization capability of our models across data sets. (ii) *Train+Test:*WSCR In the second scenario we use the predefined split of the WSCR data for training and evaluation. Since both scenarios do not involve a development set, we randomly split off 100 twin pair problems (200 examples) from the training data for development purposes. Since there is much stochasticity in the models (stochastic gradient descent, parameter sampling, training-development split, etc.), we do five random initializations with different seeds. We choose the model parameterizations from the epochs where they performed best on the development set. These models predict the test set and we compute mean and standard deviation of accuracy. We also introduce two ensembles, the naïve ensemble (NaïveE) and the Siamese ensemble (SiamE), which are majority voters informed by the predictions of the five different random seed models.

Parameter Search. We examine the Naïve model and the Siamese model, using all discussed features and pretrained, fixed 300 dimensional GloVe word embeddings (Pennington et al., 2014). Dependency edge embeddings with 10 dimensions are initialized randomly from $\mathcal{N}_{10}(0,1)$. The two embeddings for the anonymized mentions are drawn from $\mathcal{N}_{300}(0,1)$. The Bi-LSTMs have 32 hidden units each, the weight matrix used for the linear transformation of the inputs is initialized according to Glorot and Bengio (2010), who proposed this initialization scheme to bring substantially faster convergence. The weight matrix used for the linear transformation of the recurrent state is initialized as a random orthonormal matrix (Saxe et al., 2013; Mishkin and Matas, 2015) and the biases are initialized with zeros. Parameters are searched with RMSProp (learning rate 0.001) and mini-batches of size 128 over 1,000 epochs.

Results. Table 1 displays our main results in the two experiment settings, with WSCL and WSCR as testing data. Surprisingly, when we test the SOTA system of Peng et al. (2015a) on the *strict* WSCL data, the model fails to generalize. Again considering only the examples where the mention detection detected both and only both candidates and the pronoun was linked to one of them, it makes 24 correct and 22 false predictions and does not significantly outperform the random baseline (p=0.44). Our model experiences the same problem when trained on WSCR and tested on WSCL – a random process produces more or the same amount of correct predictions with p=0.14. The InferSent model, being pre-trained on large-scale NLI tasks proved to be a strong baseline and outperformed the baseline on both datasets by a notable margin, achieving the best result on WSCL (0.56 accuracy, significant on level $p<0.05$, non-significant for $p<0.005$). When trained on the WSCR training data and tested on the WSCR testing data, however our neural model significantly outperforms the random baseline by an observable margin of 9 percentage points (pp.) for Siam and 13 pp. for SiamE. A traditional coreference system and winner of the Conll 2011 Shared Task (Pradhan et al., 2011) is significantly outperformed by our neural model by 10 pp.

[8] http://cogcomp.cs.illinois.edu/page/software/_view/Winocoref

[9] Personal communication.

[10] https://github.com/facebookresearch/InferSent

	Siam		SiamE		Naïve		NaïveE		random		InferSent		SOTA	
Test	acc	p	acc	p	acc	p	acc	p	acc	p	acc	p	acc	p
WSCR	$0.59^{\pm 0.02}$	0.00	**0.63**	0.00	$0.53^{\pm 0.02}$	0.07	0.54	0.04	0.50	0.50	0.58	0.00	**0.76***	0.00
WSCL	$0.51^{\pm 0.01}$	0.30	0.54	0.13	$0.49^{\pm 0.01}$	0.50	0.51	0.38	0.50	0.50	**0.56**	0.02	$0.52^{\star}$	0.44

Table 1: Test results for different systems on two WSC data sets. $\star$ means that the score is taken from Peng et al. (2015a) (for WSCR) or was approximated by applying the published tool as described in the text (for WSCL). Underlined p-values are smaller than 0.005. Averages and standard deviations are computed over five different random initializations of Siam (and Naïve), where we averaged over those five parameterizations that performed best on the development data. p-values for Siam and Naïve are computed using the predictions of the median accuracy model determined on the development set from the the five different random initializations. All neural models use data where noun phrase candidates are anonymized.

active feature	accuracy Siam	accuracy SiamE	
word sequence only	$0.57^{\pm 0.02}$	0.59	
+ edges	$0.58^{\pm 0.01}$	0.61	sequence level
+ connotation frames	$0.58^{\pm 0.02}$	0.60	sequence level
- connotation frames	$0.59^{\pm 0.02}$	0.60	sequence level
- edges	$0.59^{\pm 0.01}$	0.61	sequence level
+ ConceptNet embedding	$0.59^{\pm 0.01}$	0.61	NP level
- ConceptNet embedding	$0.58^{\pm 0.01}$	0.59	NP level
all active	$0.59^{\pm 0.01}$	**0.63**	

Figure 4: Feature ablation experiments, where we are separately adding one of the different features to the word sequence input (+) or remove one feature from the model(-).

in accuracy when considering the ensemble model, and 6 pp. when considering the average of all five initializations with best scores on the development set (accuracy for the shared task winner was taken from (Rahman and Ng, 2012)). The naïve models fail to significantly outperform the random process strongly indicating that the Siamese ranking model is more suitable for the WSC task as it is optimized by directly learning the differences in interpretation among two highly similar proposed resolutions, one correct and one incorrect or implausible.

Anonymization. When we train and test our system on data which was *not* anonymized, the score of the Siamese ensemble model without features drops to 0.53 (p=0.059). The training loss decreased rapidly and the model exhibited little generalization capacity on unseen data. This indicates that – while neural models appear to have the potential to learn very abstract information needed for solving WSC from few data examples (561 twin pair training examples) – (i) they are very prone to overfitting when training data is scarce and not anonymized (it instantly remembers the surface noun phrases) and consequently (ii) anonymizing NPs can be very valuable for solving WSC problems, especially in a neural network setting. This is confirmed by our experiments with the neural InferSent, where the testing scores for anonymized data vs. non-anonymized data also differ observably (WSCR, non-anonymized: 0.52, anonymized: 0.58; WSCL, non-anonymized: 0.51, anonymized: 0.56).

Feature Ablations. To show the impact of individual features used in our feature-rich models Siam and SiamE, we perform experiments where we either (i) remove one specific feature from the model ('-' in Table 4) or (ii) add a single feature on top of the encoded sentence representation ('+' in Table 4). The results provide no clear picture but suggest that the complex features brought only small performance gains when applied individually, however, when applied jointly, they increase the model's performance observably from 0.59 to 0.63 accuracy. The ConceptNet NP phrase candidate level feature yields a performance increase of 2 pp. accuracy over the basic model and caused the largest drop of -4 pp. accuracy when removed from the model. On the positive side, our results suggest that non-linear neural models can learn abstract patterns based on word sequences *alone*, in contrast to successful methods from prior literature which all rely on linguistic annotation (e.g. dependency parsing) and carefully designed features and rules for accessing external knowledge bases. The Siamese model, trained *solely* on word sequences outperforms the random process significantly ($p< 0.005$).

Bill punched *Bob* in the face because **he** was being rude to Mary. *Bill* punched Bob in the face because **he** wanted to protect Mary. *John* introduced Bill because **he** knew everyone. John introduced *Bill* because **he** was new. John visited *Luke* in the hospital because **he** was sick. *John* visited Luke in the hospital because **he** lived close by.
The boss fired *the worker* when **he** stopped performing well. *The boss* fired the worker when **he** called him into the office. *The U.S.S Enterprise* tried to assist a sister ship, but **they** arrived too late to save them. The U.S.S Enterprise tried to assist *a sister ship*, but **they** did not receive help quick enough to prevent their demise. Adam failed to kill *Alexander*, so **he** hired a bodygaurd in case of a second attempt. *Adam* failed to kill Alexander, so **he** hired an assassin for the second attempt.

Figure 5: First box: Fully correctly resolved twin pair problems by all randomly initialized models. Second box: Fully falsely resolved twin pair problems by all randomly initialized models.

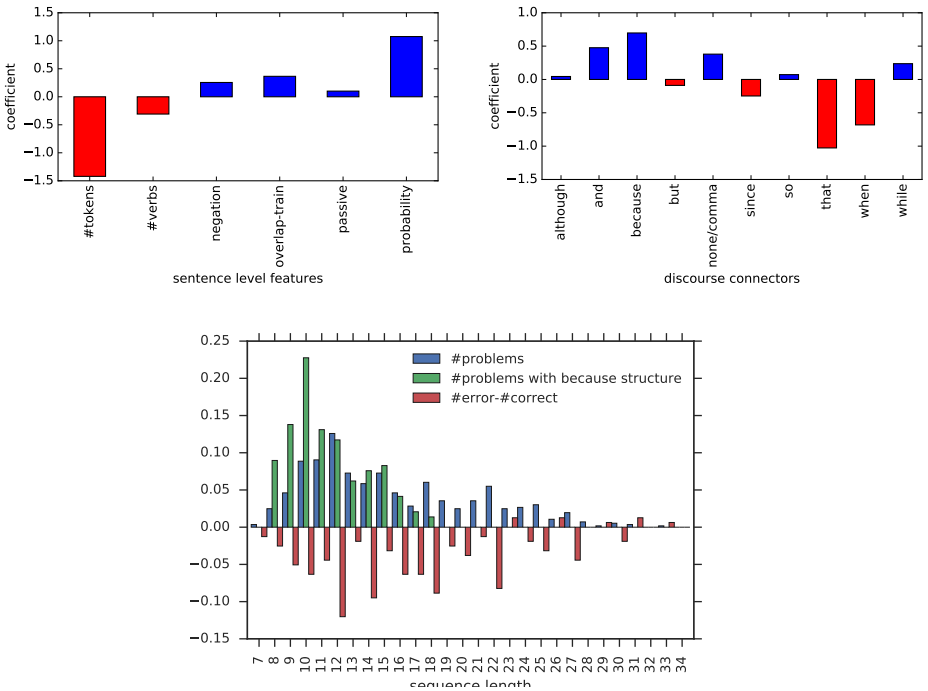

Figure 6: Coefficients for correct Siamese model guesses, sentence complexity features (left), discourse relations and *that*-conjunction (right). Bottom: Normalized distributions over sentence lengths: total (blue), problems with because-structure, amount of errors minus correct predictions. All statistics are computed from WSCR.

Deeper Analysis. In order to obtain deeper insight into the strengths and weaknesses of our model, we examine what properties discriminate the examples that our model solves successfully from the ones that it predicts erroneously. According to Levesque et al. (2012), it is critical to find a pair of twins that differ in one critical word in order to construct a full fledged WS, so it is natural that one may be interested in the model's performance over the twin pairs, i.e. the performance with respect to complete Winograd Schemata. Thereby we may also gain a better intuition of how vulnerable the model is with regard to changing the critical word. Figure 5 displays twin sentences from WSCR, where all five randomly initialized models came to the made the same prediction over the whole pair. Complete twin pairs were resolved correctly in 10 cases and incorrectly in 7 cases. The first case can be seen as 'most easy' for the model while we can conclude that the second case appeared to be the 'most difficult' or 'confusing' cases for the model. The examples suggest that the models perform better on sentences with unambiguous causal discourse markers (*because*) and less linguistic complexity (less verbs, shorter in length). To investigate more closely to what extent a successful resolution is informed by linguistic complexity, we designed 6 linguistic sentence-level features (length, number of verbs, passive construction, negation, sequence probability estimated with a language model and ratio of tokens to be found in the training data)

and 10 binary features for different discourse connectors (*because, when, while*, etc.) and the sentence embedding conjunction *that*. From all five Siamese model initializations we collect the predictions, normalize the features onto a range between 0 and 1 and fit a regularized logistic regression model to predict a correct or incorrect prediction based on the aforementioned features. The coefficients of the features are displayed in Figure 6. The sequence length is strongly negatively correlated with a successful model prediction. On the other hand, the higher the estimated sentence probability and overlap with the training data, the more likely the Siamese model is to make a correct prediction. Perhaps more interesting are the coefficients for the discourse relation features. As the examples in (Figure 5) already suggested, the Siamese model performs better with the unambiguous causal discourse connector *because* as opposed to the ambiguous connector *when* or the sentence embedding conjunction *that*. However, this can also be explained by the fact that *because* is the most common discourse marker in the training data (698 occurrences in 1322 problems). Also, we found that problems involving *because* are generally shorter than other sentences in the data (see Figure 6, bottom).

6 Conclusion

Our assumption is that for interpreting Winograd sentences, humans process and build up a representation for full sentences, and that based on their understanding of the sentence with one or the other way of resolving the pronominal reference, they are able to decide which reading is correct. How exactly this is performed in terms of cognitive processes we cannot answer. However, the approach we are proposing offers two important ingredients of such a potential/hypothesized interpretation process: we formalized the WSC as a general sequence ranking problem and designed a Siamese neural network model that (i) computes full-fledged sentence interpretations as they would emerge from from resolving the pronominal anaphor to one or the other antecedent , and (ii) a ranking function that decides which of these interpretations can be assigned a higher confidence. Our Siamese model is able to solve a considerable amount of WSC challenge questions, after training it on pairs of sentence representations with correctly vs. incorrectly resolved anaphoric pronouns, where it learns information (features) that distinguishes these pairs. When applying the learned model to unseen pairs, it significantly outperforms not only a random process but also a naïve baseline neural model. While the model still lags behind state-of-the-art linear systems that rely on syntactic preprocessing and complex external knowledge sources accessed by manually designed features, our results are most promising: the Siamese sequence ranking model is able to learn how to resolve WS by only considering word sequences as input, and does so significantly better than the random baseline.

Cross-dataset experiments however showed that the WSC is far from being solved – while a state-of-the-art method and our system successfully answer many problems in one testing set (where the training data stems from the same source, created by a class of undergraduate students), *both* fail to generalize when presented a different, smaller WSC data set (where the examples perhaps are more carefully designed and seem notably more natural). On the smaller data both systems do not significantly outperform a random process. Because of this drastic drop in all of the model's performances and the small amounts of data we suggest that future work on the WSC should carefully test the methods on as much data as is available.

Our task formulation provides an easily accessible way for other researchers working on textual understanding to quickly test their sentence models on a very important AI and text understanding task.

Acknowledgements

This work has been supported by the German Research Foundation as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1 and by the Leibniz ScienceCampus "Empirical Linguistics and Computational Language Modeling", supported by the Leibniz Association grant no. SAS-2015-IDS-LWC and by the Ministry of Science, Research, and Art of Baden-Württemberg.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chitta Baral. 2003. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, New York, NY, USA.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Thorsten Brants and Alex Franz, 2006. *Web 1T 5-gram Version 1. Linguistic Data Consortium, Philadelphia, PA*. Philadelphia, PA.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio, June. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. *CoRR*, abs/1705.02364.

Michael Gelfond and Vladimir Lifschitz. 1988. The Stable Model Semantics For Logic Programming. pages 1070–1080. MIT Press.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-Normalizing Neural Networks. *CoRR*, abs/1706.02515.

Canasai Kruengkrai, Naoya Inoue, Jun Sugiura, and Kentaro Inui. 2014. An Example-Based Approach to Difficult Pronoun Resolution. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 358–367, Phuket,Thailand, December. Department of Linguistics, Chulalongkorn University.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's Multi-pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. *CoRR*, abs/1707.07045.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press.

Hector J. Levesque. 2014. On our best behaviour. *Artificial Intelligence*, 212:27 – 35.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November.

Dmytro Mishkin and Jiri Matas. 2015. All you need is a good init. *CoRR*, abs/1511.06422.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015a. A Joint Framework for Coreference Resolution and Mention Head Detection. In *CoNLL*, page 10, University of Illinois, Urbana-Champaign, Urbana, IL, 61801, 7. ACL.

Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015b. Solving Hard Coreference Problems. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 809–819. The Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 1–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, pages 777–789.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2015. Connotation Frames: Typed Relations of Implied Sentiment in Predicate-Argument Structure. *CoRR*, abs/1506.02739.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation Frames: A Data-Driven Investigation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120.

Alexander Schrijver. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA.

Peter Schüller. 2014. Tackling Winograd Schemas by Formalizing Relevance Theory in Knowledge Graphs. In *KR*. AAAI Press.

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. AAAI Press.

Robert Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5.

Robert Speer and Joanna Lowry-Duda. 2017. ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge. *CoRR*, abs/1704.03560.