Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning
Main Track - Short paper

PATCHCOMM: Using Commonsense Knowledge to Guide Syntactic Parsers

Yida Xin[1] , **Henry Lieberman**[2] , **Peter Chin**[1]
[1]Boston University
[2]Massachusetts Institute of Technology
yxin@bu.edu, lieber@media.mit.edu, spchin@cs.bu.edu

Abstract

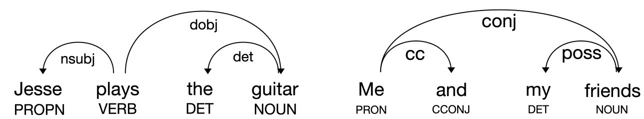
Syntactic parsing technologies have become significantly more robust thanks to advancements in their underlying statistical and Deep Neural Network (DNN) techniques: most modern syntactic parsers can produce a syntactic parse tree for almost any sentence, including ones that may not be strictly grammatical. Despite improved robustness, such parsers still do not reflect the alternatives in parsing that are intrinsic in syntactic ambiguities. Two most notable such ambiguities are prepositional phrase (PP) attachment ambiguities and pronoun coreference ambiguities. In this paper, we discuss PATCHCOMM, which uses commonsense knowledge to help resolve both kinds of ambiguities. To the best of our knowledge, we are the first to propose the general-purpose approach of using external commonsense knowledge bases to guide syntactic parsers. We evaluated PATCHCOMM against the state-of-the-art (SOTA) spaCy parser on a PP attachment task and against the SOTA NeuralCoref module on a coreference task. Results show that PATCHCOMM is successful at detecting syntactic ambiguities and using commonsense knowledge to help resolve them.

1 Introduction

The central claim for this paper is that syntactic parsing is a natural application and test bed for commonsense reasoning.

In this section starting with the next paragraph, we introduce the basic notion of syntactic parsing. In section 2, we explain PATCHCOMM, a commonsense-based approach to tackling syntactic ambiguities in parsing. In section 3, we show experimental results that provide evidence for our central claim. In section 4, we survey some related work on commonsense and knowledge bases, as well as the two prevalent ambiguities in syntactic parsing that we tackle in section 2. In section 5, we discuss future directions.

Syntactic parsing is the linguistic process of analyzing the grammatical and syntactic information of a sentence and compiling such information into a *syntactic parse tree*. Two of the most popular paradigms for syntactic parsing are *dependency* and *constituency*. Especially over the last decade, syntactic parsing technologies have become significantly more robust: Regardless of how ungrammatical a sentence might be, a parser can be expected to produce a parse tree for that sentence. In Figure 1, we provide examples of syntactic parse trees produced by the parser of a state-of-the-art (SOTA) and popular natural language processing (NLP)



(a) spaCy dependency parse of "Jesse plays the guitar."

(b) spaCy dependency parse of "Me and my friends."

Figure 1: Examples of syntactic parses.

toolkit, spaCy (Honnibal et al. 2020). Note that in Figures 1 (a) and (b), both parsers gave outputs to the ungrammatical/incomplete sentence, "Me and my friends," with correct syntactic relations.

Despite this robustness, challenging linguistic ambiguities abound. For the remainder of this paper, we focus on and posit solutions to two such ambiguities – prepositional phrase (PP) attachment ambiguities and pronoun coreference ambiguities. Because the coreference problem is less constrained by linguistic structure and more constrained by the underlying contextual and commonsense knowledge (section 4.1), it is arguable that coreference resolution can be more challenging.

2 Related Work

2.1 Syntactic Parsing Exposes the Need for Underlying Commonsense Knowledge

The examples in Figure 3 (which we explain in Section 3) demonstrate why parsers need to be helped with underlying commonsense knowledge in order to parse "better." Namely, in those examples, the parser needs to know or be told that people have arms but not wings, birds have wings but not arms, ice is generally hard and butter is generally soft.

PP Attachment Ambiguities (Hamdan and Khan 2018), in addition to enumerating an ontology of linguistic ambiguities, comprehensively surveys what they refer to, in their section 3, as *earlier*, *corpus-based* and *statistical* approaches to resolving sentence-level PP attachment ambiguities, but these three approaches are not mutually exclusive. Of all the surveyed works, we first highlight (Ratnaparkhi, Reynar, and Roukos 1994); to the best of our knowledge, they were the first to have proposed the syntactic constraint of (V, N_1, P, N_2), where V is *verb*, N_1 and N_2 are *nouns*, P

is *preposition*, and together (P, N_2) is the PP. In addition, we note that the corpus-based and statistical approaches have been successful at unraveling and taking advantage of underlying biases intrinsic to specific corpora and prepositions. For example, (Cimiano and Minock 2010) discovers that, for the 883 test questions from the GeoBase dataset[1], always attaching PPs and relative clauses to the last constituent[2] gave rise to a baseline of 99.27% accuracy. This hack would not generally work well for non-queries. Similarly, (Bailey, Lierler, and Susman 2015) reiterates that, 99% of the times, an "of" PP would be attached to N_1 instead of V.

Pronoun Coreference Ambiguities In light of coreference ambiguities, (Kocijan et al. 2020) comprehensively surveys a recently very hot series of works on tackling a famous instance of coreference ambiguities: *Winograd Schemas*. In its basic form, a Winograd Schema (WS) (Winograd 1972) is a pair of sentences that look like the pairs of sentences in Figure 3 for NeuralCoref, where there is one ambiguous pronoun and two distinct entities, i.e., $(\cdots Entity_1 \cdots Entity_2 \cdots pronoun \cdots)$. Changing the description word (e.g., *hard*) to a word of the opposite meaning (e.g., *soft*) leads to changing the pronoun's (e.g., *it*) reference from one entity (e.g., *ice*) to the other (e.g., *butter*).

The series of works started with (Levesque, Davis, and Morgenstern 2012), which proposed the original Winograd Schema Challenge (WSC) as an alternative to Turing Test because, as Levesque et al. claimed, commonsense reasoning is a must for resolving WS. The series of works culminated in (Sakaguchi et al. 2020), which proposed the large-scale crowdsourced dataset, WINOGRANDE, accompanied by a dataset debiasing algorithm, AFLITE. The authors then showed that transfer learning (Yosinski et al. 2014) from WINOGRANDE to other WSC-related datasets boosts performances on those datasets, including a human-level performance of 93.1%. However, in a footnote, the authors admitted that SOTA models that were trained on the AFLITE-debiased version of WINOGRANDE "showed only chance level performance." It is open to debate how much commonsense such WSC-inspired DNN models in fact exhibit.

2.2 Commonsense Knowledge

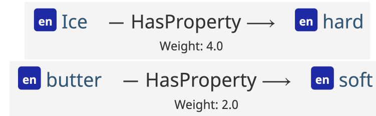
Although there is no unified definition of *commonsense*, many have attempted at their own definitions that capture at least aspects of commonsense. One such attempt is in (Mueller 2014), which treats *commonsense reasoning* as a logic-based process that "[takes] information about certain aspects of a scenario in the world and [makes] inferences about other aspects of the scenario," and defines *commonsense knowledge* as the prerequisite knowledge for commonsense reasoning.

Another school of thought treats commonsense as an emergent property of intricate interactions among "societies" of specialized software (Minsky 1986; Minsky 2006).

[1]Dataset: https://www.cs.utexas.edu/users/ml/nldata.html.

[2]For example, for the query "How many states in the U.S. does the shortest river run through?," the PP, *in the U.S.*, would by default be attached to the last constituent, *states*.

(a) ConceptNet knows people have arms and birds have wings.



(b) ConceptNet knows ice is hard and butter is soft.

Figure 2: Examples of commonsense assertions in ConceptNet.

By either definition, PATCHCOMM can be thought of as an instance of commonsense reasoning, where a sentence is a description of some scenario, and a parser (i.e., specialized software) takes syntactic information about a sentence and, with the help of a CSKB (i.e., another specialized software), makes inferences about such information.

One notable effort in collecting commonsense knowledge was the Open Mind Common Sense (OMCS) project (Singh et al. 2002), which later conceived the ConceptNet (Speer, Chin, and Havasi 2017) knowledge graph. Since then, there have been other CSKBs resembling ConceptNet in spirit; one example is WebChild (Tandon et al. 2014; Tandon, De Melo, and Weikum 2017). At the time of this writing, ConceptNet is in its version 5.8. At its core, ConceptNet is a very-large-scale, directed graph whose nodes represent *concepts* and directed, labeled edges represent *relations* that relate one concept to another. Each such (*concept1*, *relation*, *concept2*) triple in ConceptNet is called an *assertion*. Figure 2 shows that ConceptNet knows people have arms, birds have wings, ice is hard, and butter is soft.

Another notable effort in collecting large-scale commonsense knowledge bases is ATOMIC (Sap et al. 2019), whose nodes represent *events*, *mental states*, or *personas*, and directed edges represent *If-Then relations* that relate an event to a mental state, a persona, or another event.

3 PATCHCOMM

In this section, we discuss how PATCHCOMM guides the spaCy parser[3] on resolving PP attachment ambiguities and compares with the SOTA, Deep Neural Network (DNN) based NeuralCoref[4] module[5] (Clark and Manning 2016b; Clark and Manning 2016a) on resolving coreference ambiguities.

Contrasting with many approaches that we survey in Section 2, PATCHCOMM is a general-purpose framework that modularizes syntactic parsing into a preliminary parser and a commonsense knowledge base (CSKB). The CSKB serves as a "critic" (Minsky 2006) that provides guidance for the parser's preliminary outputs.

[3]We used spaCy version 2.3.5.

[4]Source: https://github.com/huggingface/neuralcoref

[5]We used NeuralCoref version 4.1.0.

Although all are knowledge-based approaches, PATCHCOMM is distinguished from (Belinkov et al. 2014) and (Nakashole and Mitchell 2015) in two main ways: (1) The knowledge we use are commonsense plus linguistic knowledge (because ConceptNet 5 contains both) rather than linguistic knowledge alone; (2) instead of using knowledge internally to the parser to help generate feature vectors and then treat the PP attachment problem as binary or multi-class classification, we modularize knowledge into an external *guiding* module for the parser. That being said, we concur that using knowledge both internally and externally is a plausible approach for future work.

In its current version, in addition to spaCy and NeuralCoref, PATCHCOMM uses the very-large-scale ConceptNet[6] (Speer, Chin, and Havasi 2017) for the CSKB. See section 2.2 for details on ConceptNet.

3.1 System Details

PATCHCOMM makes full use of spaCy's built-in parsing mechanisms, including those for obtaining part-of-speech tags, syntactic dependency labels, and child-head relations.

One Ambiguity per Sentence Abiding by the (V, N_1, P, N_2) constraint for PP attachment (section 2.1.1), PATCHCOMM first checks whether spaCy has attached the PP (i.e., (P, N_2)) to V or N_1, and then queries ConceptNet with the pairs ($subj$, N_2), (V, N_2) and (N_1, N_2) where $subj$ is the subject of V. Oftentimes in a sentence, $subj$ contains useful information about V, therefore PATCHCOMM checks for knowledge about the subject, too. If ConceptNet indicates a stronger connection for ($subj$, N_2) or (V, N_2) than for (N_1, N_2), PATCHCOMM informs spaCy to attach the PP to V; otherwise, PATCHCOMM informs spaCy to attach the PP to N_2.

When queried with a pair of concepts, (c_1, c_2), ConceptNet returns all relations that start with either c_1 or c_2 and end with either c_2 or c_1. Because ConceptNet is not omniscient, whenever PATCHCOMM cannot find assertions (section 4.1) from ConceptNet, it defaults back to spaCy's attachment decisions. But when successful at finding assertions, PATCHCOMM takes the assertion with the highest weight and goes on to process the next pair of concepts. Whenever there is a mismatch between the CSKB's knowledge and the parser's output, PATCHCOMM abides by the CSKB's knowledge and modifies the parser's output accordingly.

For coreference resolution, PATCHCOMM obeys the constraint of $(\cdots Entity_1 \cdots Entity_2 \cdots pronoun \cdots)$ (section 2.1.2). PATCHCOMM finds the *description* of the pronoun in the sentence, which is defined as the token that shares the same head with the pronoun. Then, PATCHCOMM queries ConceptNet with the pairs ($Entity_1$, $description$) and ($Entity_2$, $description$).

Multiple Ambiguities per Sentence PATCHCOMM's mechanism even works in the cases of multiple PP attachment ambiguities and multiple coreference ambiguities, respectively.

[6]We used ConceptNet version 5.8; see http://blog.conceptnet.io/.

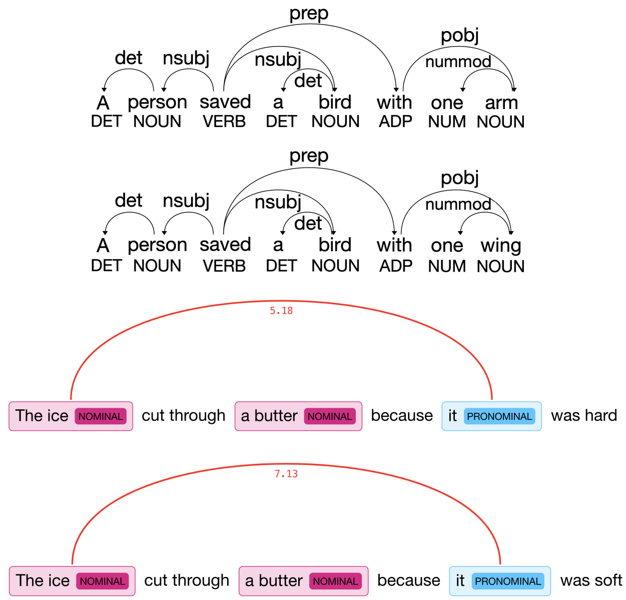


Figure 3: spaCy (top) and NeuralCoref (bottom) when faced with PP attachment and coreference ambiguities, respectively.

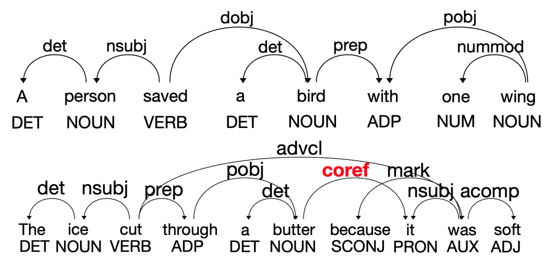


Figure 4: PATCHCOMM resolves ambiguities from Figure 3.

For PP attachment, for a sentence with $n (n \geq 2)$ PPs and for $i \in [1, \cdots, n]$, every time after PATCHCOMM resolves the i-th PP and moves onto the $(i+1)$-th PP, PATCHCOMM now treats all of (V, N_1, $\{pobj_j\}_1^i$) as candidates for attachment, where $pobj_j$ is the *object of preposition* for the j-th PP, where $j \in [1, \cdots, i]$.

For coreference, PATCHCOMM assumes that each pronoun can refer to any entity that occurs in the sentence, and uses the same description-token-matching mechanism, which is detailed in Section 3.1.1.

3.2 Working Examples

Figure 3 shows examples of PP attachments performed by spaCy and coreference resolutions performed by NeuralCoref without PATCHCOMM. Figure 4 shows how PATCHCOMM helps improve some outputs in Figure 3. Note in Figure 4 that PATCHCOMM embeds coreference information into spaCy-style dependency parse, as shown by the red highlighted *coref* label.

In addition, Figure 5 compares spaCy and PATCHCOMM on resolving the same sentence for multiple PPs.

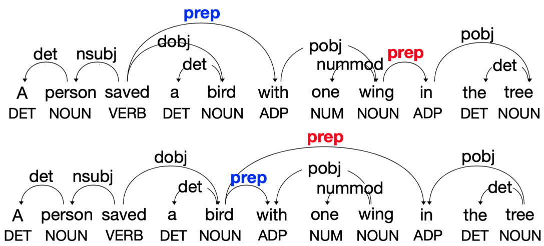


Figure 5: spaCy (top) and PATCHCOMM (bottom) resolving PP attachments in a sentence; first PP in blue, second PP in red.

3.3 Some Future Work for PATCHCOMM

There are two immediate next steps for PATCHCOMM:

For the linguistic step, PATCHCOMM is underway for resolving multiple ambiguous pronouns, as well as PP attachment ambiguities and coreference ambiguities combined.

For the knowledge step, PATCHCOMM is underway for incorporating inference modules that embed ConceptNet into DNN-based systems and querying these systems instead of ConceptNet, in order to make up for ConceptNet's sparse and incomplete coverage. This approach resembles (Bosselut et al. 2019) in spirit.

4 Experiments and Results

First, we summarize the most important result of our experiments, in Table 1. Then, we explain the details of how the experiments were performed. The bottom line is that our system provided significant improvements over baselines, especially for the coreference task.

	Baseline	PATCHCOMM
PP Attachment	57%	**61%**
Coreference	30.77%	**51.65%**

Table 1: Baselines are spaCy (top) and NeuralCoref (bottom).

4.1 PP Attachment

We created a small dataset that is partially based on that of (Belinkov et al. 2014)[7], with partial help from *Write With Transformer*[8]. The dataset has 100 sentences, with distribution of prepositions shown in Figure 6. The Y-axis is the number of instances of that preposition in our 100-sentence dataset. For baseline, we used spaCy (Honnibal et al. 2020).

Out of these 100 sentences, spaCy scored 57 correct by itself and 61 correct using PATCHCOMM. There are 8 sentences for which spaCy made different PP attachments than PATCHCOMM, suggesting that (1) PATCHCOMM needs to make fuller user of a sentence's contextual information; (2) PATCHCOMM also needs to use commonsense knowledge in a *context-dependent* manner.

[7]Belinkov et al. dataset available: https://github.com/boknilev/pp-attachment/tree/master/data/pp-data-english

[8]https://transformer.huggingface.co/

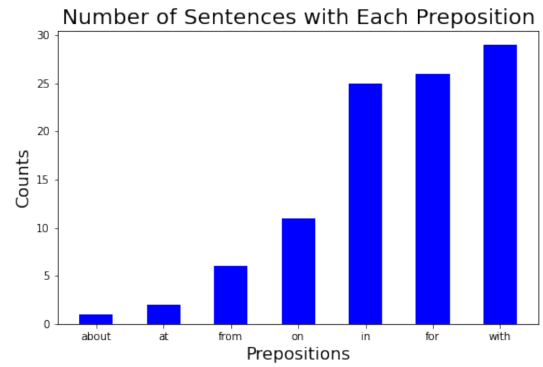


Figure 6: Distribution of prepositions in our PP attachment dataset.

4.2 Coreference Resolution

For coreference, we used the WSC273 dataset[9] (Levesque, Davis, and Morgenstern 2012). This dataset has 273 sentences of the format $(\cdots Entity_1 \cdots Entity_2 \cdots pronoun \cdots)$. For baseline, we used NeuralCoref (Clark and Manning 2016b; Clark and Manning 2016a).

Out of these 273 sentences, NeuralCoref scored 30.77% correct and PatchComm scored 51.65% correct. Very curiously, the set of sentences scored by NeuralCoref was completely disjoint from the set of sentences scored by PATCHCOMM. We suspect this might be because NeuralCoref has a curious capability of overlooking "obvious" commonsense that is naturally captured by CSKBs such as ConceptNet, but capturing certain "obscure" commonsense that is not immediately made clear in CSKBs.

5 Conclusion

We first introduced our vision that syntactic parsing is a natural application and test bed for commonsense reasoning. To showcase this, we pointed out that syntactic parsers need considerable commonsense knowledge to make good parsing decisions, and that such knowledge can either stem internally from the parsers or come externally from CSKBs. To showcase that external CSKBs do help, we introduced PATCHCOMM, which uses an external CSKB, ConceptNet, to provide commonsense knowledge for a parser, spaCy. Our results encourage our vision.

We are also reminded of the necessity and urgency to integrate the research communities for knowledge representation and language understanding. Knowledge is an indispensable element for language understanding; language is an essential conveyance of knowledge, at least for people. We believe this collaboration is necessary for building Artificial Intelligence (AI) systems that possess true communication capacity, which in turn is the foundation for AI transparency, explainability, fairness, and safety.

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

References

Bailey, D.; Lierler, Y.; and Susman, B. 2015. Prepositional phrase attachment problem revisited: how verbnet can help. In *Proceedings of the 11th International Conference on Computational Semantics*, 12–22. London, UK: Association for Computational Linguistics.

Belinkov, Y.; Lei, T.; Barzilay, R.; and Globerson, A. 2014. Exploring compositional architectures and word vector representations for prepositional phrase attachment. *Transactions of the Association for Computational Linguistics* 2:561–572.

Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Celikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4762–4779. Florence, Italy: Association for Computational Linguistics.

Cimiano, P., and Minock, M. 2010. Natural language interfaces: What is the problem? – a data-driven quantitative analysis. In Horacek, H.; Métais, E.; Muñoz, R.; and Wolska, M., eds., *Natural Language Processing and Information Systems*, 192–206. Berlin, Heidelberg: Springer Berlin Heidelberg.

Clark, K., and Manning, C. D. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2256–2262. Austin, Texas: Association for Computational Linguistics.

Clark, K., and Manning, C. D. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 643–653. Berlin, Germany: Association for Computational Linguistics.

Hamdan, M., and Khan, I. 2018. An analysis of prepositional-phrase attachment disambiguation. *International Journal of Computational Linguistics Research* 9:60.

Honnibal, M.; Montani, I.; Van Landeghem, S.; and Boyd, A. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Kocijan, V.; Lukasiewicz, T.; Davis, E.; Marcus, G.; and Morgenstern, L. 2020. A review of winograd schema challenge datasets and approaches. *arXiv preprint arXiv:2004.13831 [cs.CL]*.

Levesque, H.; Davis, E.; and Morgenstern, L. 2012. The winograd schema challenge. In *13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012*, Proceedings of the International Conference on Knowledge Representation and Reasoning, 552–561. Institute of Electrical and Electronics Engineers Inc. 13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012 ; Conference date: 10-06-2012 Through 14-06-2012.

Minsky, M. 1986. *The Society of Mind*. USA: Simon Schuster, Inc.

Minsky, M. 2006. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. USA: Simon Schuster, Inc.

Mueller, E. T. 2014. *Commonsense Reasoning: An Event Calculus Based Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2 edition.

Nakashole, N., and Mitchell, T. M. 2015. A knowledge-intensive model for prepositional phrase attachment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 365–375. Beijing, China: Association for Computational Linguistics.

Ratnaparkhi, A.; Reynar, J.; and Roukos, S. 1994. A maximum entropy model for prepositional phrase attachment. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Sakaguchi, K.; Le Bras, R.; Bhagavatula, C.; and Choi, Y. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(05):8732–8740.

Sap, M.; Le Bras, R.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N. A.; and Choi, Y. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3027–3035.

Singh, P.; Lin, T.; Mueller, E. T.; Lim, G.; Perkins, T.; and Li Zhu, W. 2002. Open mind common sense: Knowledge acquisition from the general public. In Meersman, R., and Tari, Z., eds., *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, 1223–1237. Berlin, Heidelberg: Springer Berlin Heidelberg.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence* 31(1).

Tandon, N.; De Melo, G.; Suchanek, F.; and Weikum, G. 2014. Webchild: Harvesting and organizing commonsense knowledge from the web. In *Proceedings of the 7th ACM international conference on Web search and data mining*, 523–532.

Tandon, N.; De Melo, G.; and Weikum, G. 2017. Webchild 2.0: Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations*, 115–120.

Winograd, T. 1972. Understanding natural language. *Cognitive Psychology* 3(1):1–191.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.