Machine Comprehension using Rich Semantic Representations

Mrinmaya Sachan **Eric P. Xing**
School of Computer Science
Carnegie Mellon University
{mrinmays, epxing}@cs.cmu.edu

Abstract

Machine comprehension tests the system's ability to understand a piece of text through a reading comprehension task. For this task, we propose an approach using the Abstract Meaning Representation (AMR) formalism. We construct meaning representation graphs for the given text and for each question-answer pair by merging the AMRs of comprising sentences using cross-sentential phenomena such as coreference and rhetorical structures. Then, we reduce machine comprehension to a graph containment problem. We posit that there is a latent mapping of the question-answer meaning representation graph onto the text meaning representation graph that explains the answer. We present a unified max-margin framework that learns to find this mapping (given a corpus of texts and question-answer pairs), and uses what it learns to answer questions on novel texts. We show that this approach leads to state of the art results on the task.

1 Introduction

Learning to efficiently represent and reason with natural language is a fundamental yet long-standing goal in NLP. This has led to a series of efforts in broad-coverage semantic representation (or "sembanking"). Recently, AMR, a new semantic representation in standard neo-Davidsonian (Davidson, 1969; Parsons, 1990) framework has been proposed. AMRs are rooted, labeled graphs which incorporate PropBank style semantic roles, within-sentence coreference, named entities and the notion of types, modality, negation, quantification, etc. in one framework.

In this paper, we describe an approach to use AMR for the task of machine comprehension. Machine comprehension (Richardson et al., 2013) evaluates a machine's *understanding* by posing a series of multiple choice reading comprehension tests. The tests are unique as the answer to each question can be found only in its associated texts, requiring us to go beyond simple lexical solutions. Our approach models machine comprehension as an extension to textual entailment, learning to output an answer that is best *entailed* by the passage. It works in two stages. First, we construct a meaning representation graph for the entire passage (§ 2.1) from the AMR graphs of comprising sentences. To do this, we account for cross-sentence linguistic phenomena such as entity and

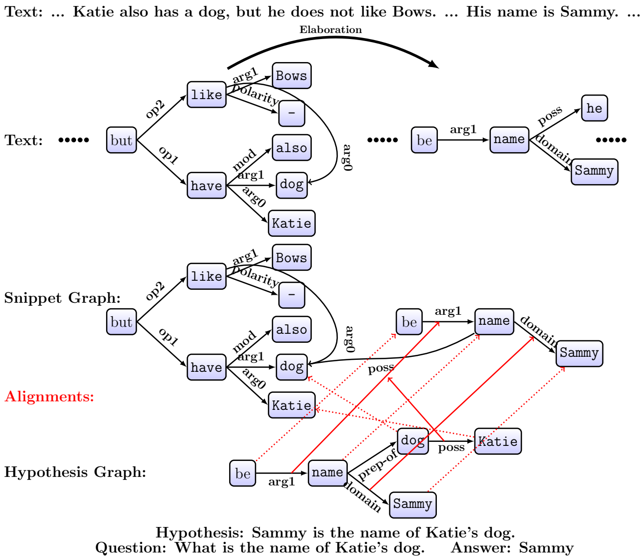
Text: ... Katie also has a dog, but he does not like Bows. ... His name is Sammy. ...
Elaboration
Text:
but
op2
op1
like
arg1
polarity
Bows
-
have
mod
arg1
arg0
also
dog
Katie
arg0
be
arg1
name
poss
domain
he
Sammy
Snippet Graph:
Alignments:
Hypothesis Graph:
prep-of
poss
Hypothesis: Sammy is the name of Katie's dog.
Question: What is the name of Katie's dog. Answer: Sammy

Figure 1: Example latent *answer-entailing* structure from the MCTest dataset. The question and answer candidate are combined to generate a hypothesis. This hypothesis is AMR parsed to construct a hypothesis meaning representation graph after some post-processing (§ 2.1). Similar processing is done for each sentence in the passage as well. Then, a subset (not necessarily contiguous) of these sentence meaning representation graphs is found. These representation subgraphs are further merged using coreference information, resulting into a structure called the relevant text snippet graph. Finally, the hypothesis meaning representation graph is aligned to the snippet graph. The dashed red lines show node alignments, solid red lines show edge alignments, and thick solid black arrow shows the rhetorical structure label (elaboration).

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 486–492,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

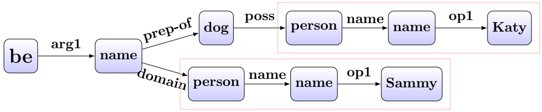

Figure 2: The AMR parse for the hypothesis in Figure 1. The person nodes are merged to achieve the hypothesis meaning representation graph.

event coreference, and rhetorical structures. A similar meaning representation graph is also constructed for each question-answer pair. Once we have these graphs, the comprehension task henceforth can be reduced to a graph containment problem. We posit that there is a latent subgraph of the text meaning representation graph (called snippet graph) and a latent alignment of the question-answer graph onto this snippet graph that *entails* the answer (see Figure 1 for an example). Then, we propose a unified max-margin approach (§ 2.2) that jointly learns the latent structure (subgraph selection and alignment) and the QA model. We evaluate our approach on the MCTest dataset and achieve competitive or better results than a number of previous proposals for this task.

2 The Approach

2.1 The Meaning Representation Graphs

We construct the meaning representation graph using individual sentences AMR graphs and merging identical concepts (using entity and event coreference). First, for each sentence AMR, we merge nodes corresponding to multi-word expressions and nodes headed by a date entity ("date-entity"), or a named entity ("name") or a person entity ("person"). For example, the hypothesis meaning representation graph in Figure 1 was achieved by merging the AMR parse shown in Figure 2.

Next, we select the subset of sentence AMRs corresponding to sentences needed to answer the question. This step uses cross-sentential phenomena such as rhetorical structures[1] and entities/event coreference. The coreferent entities/event mentions are further merged into one node resulting in a graph called the relevant text snippet graph. A similar process is also performed with the hypothesis sentences (generated by combining the question and answer candidate) as shown in Figure 1.

2.2 Max-Margin Solution

For each question $q_i \in Q$, let $\mathbf{t}_i$ be the corresponding passage text and $A_i = \{a_{i1}, \ldots, a_{im}\}$ be the set of candidate answers to the question. Our solution casts the machine comprehension task as a textual entailment task by converting each question-answer candidate pair (q_i, a_{ij}) into a hypothesis statement h_{ij}. We use the question matching/rewriting rules described in Cucerzan and Agichtein (2005) to get the hypothesis statements. For each question q_i, the machine comprehension task reduces to picking the hypothesis $\hat{h}_i$ that has the highest likelihood of being entailed by the text $\mathbf{t}_i$ among the set of hypotheses $\mathbf{h}_i = \{h_{i1}, \ldots, h_{im}\}$ generated for the question q_i. Let $h_i^* \in \mathbf{h}_i$ be the hypothesis corresponding to the correct answer.

As described, we use subgraph matching to help us model the inference. We assume that the selection of sentences to generate the relevant text snippet graph and the mapping of the hypothesis meaning representation graph onto the passage meaning representation graph is latent and infer it jointly along with the answer. We treat it as a structured prediction problem of ranking the hypothesis set $\mathbf{h}_i$ such that the correct hypothesis h_i^* is at the top of this ranking. We learn a scoring function $S_{\mathbf{w}}(\mathbf{t}, h, \mathbf{z})$ with parameter $\mathbf{w}$ such that the score of the correct hypothesis h_i^* and corresponding best latent structure $\mathbf{z}_i^*$ is higher than the score of the other hypotheses and corresponding best latent structures. In a max-margin fashion, we want that $S_{\mathbf{w}}(\mathbf{t}_i, h_i^*, \mathbf{z}_i^*) > S(\mathbf{t}_i, h_{ij}, \mathbf{z}_{ij}) + 1 - \xi_i$ for all $h_j \in \mathbf{h} \setminus h^*$ for some slack ξ_i. Writing the relaxed max margin formulation:

$$\min_{||\mathbf{w}||} \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_i \max_{\mathbf{z}_{ij}, h_{ij} \in \mathbf{h}_i \setminus h_i^*} S_{\mathbf{w}}(\mathbf{t}_i, h_{ij}, \mathbf{z}_{ij}) + \Delta(h_i^*, h_{ij})$$
$$-C \sum_i S_{\mathbf{w}}(\mathbf{t}_i, h_i^*, \mathbf{z}_i^*) \qquad (1)$$

We use 0-1 cost, i.e. $\Delta(h_i^*, h_{ij}) = \mathbb{1}(h_i^* \neq h_{ij})$. If the scoring function is convex then this objective is in concave-convex form and hence can be solved by the concave-convex programming procedure (CCCP) (Yuille and Rangarajan, 2003). We assume the scoring function to be linear:$S_{\mathbf{w}}(\mathbf{t}, h, \mathbf{z}) = \mathbf{w}^T \psi(\mathbf{t}, h, \mathbf{z})$. Here,

[1] Rhetorical structure theory (Mann and Thompson, 1988) tells us that sentences with discourse relations are related to each other. Previous works in QA (Jansen et al., 2014) have shown that these relations can help us answer certain kinds of questions. As an example, the "cause" relation between sentences in the text can often give cues that can help us answer "why" or "how" questions. Hence, the passage meaning representation also remembers RST relations between sentences.

$\psi(\mathbf{t}, h, \mathbf{z})$ is a feature map discussed later. The CCCP algorithm essentially alternates between solving for $\mathbf{z}_i^*$, $\mathbf{z}_{ij}$ $\forall j$ s.t. $h_{ij} \in \mathbf{h}_i \setminus h_i^*$ and $\mathbf{w}$ to achieve a local minima. In the absence of information regarding the latent structure $\mathbf{z}$ we pick the structure that gives the best score for a given hypothesis i.e. $\arg\max_z S_{\mathbf{w}}(\mathbf{t}, h, z)$.

2.3 Scoring Function and Inference

Now, we define the scoring function $S_{\mathbf{w}}(\mathbf{t}, h, \mathbf{z})$. Let the hypothesis meaning representation graph be $G' = (V', E')$. Our latent structure $\mathbf{z}$ decomposes into the selection ($\mathbf{z}_s$) of relevant sentences that lead to the text snippet graph G, and the mapping ($\mathbf{z}_m$) of every node and edge in G' onto G. We define the score such that it factorizes over the nodes and edges in G'. The weight vector $\mathbf{w}$ also has three components $\mathbf{w}_s$, $\mathbf{w}_v$ and $\mathbf{w}_e$ corresponding to the relevant sentences selection, node matches and edge matches respectively. An edge in the graph is represented as a triple (v^1, r, v^2) consisting of the enpoint vertices and relation r.

$$\begin{aligned} S_{\mathbf{w}}(\mathbf{t}, h, \mathbf{z}) = {} & \mathbf{w}_s^T \mathbf{f}(G', G, \mathbf{t}, h, \mathbf{z}_s) \\ & + \sum_{v' \in V'} \mathbf{w}_v^T \mathbf{f}(v', z_m(v')) + \sum_{e' \in E'} \mathbf{w}_e^T \mathbf{f}(e', z_m(e')) \end{aligned}$$

Here, $\mathbf{t}$ is the text corresponding to the hypothesis h, and $\mathbf{f}$ are parts of the feature map ψ to be described later. $z(v')$ maps a node $v' \in V'$ to a node in V. Similarly, $z(e')$ maps an edge $e' \in E'$ to an edge in E.

Next, we describe the inference procedure i.e. how to select the structure that gives the best score for a given hypothesis. The inference is performed in two steps: The first step selects the relevant sentences from the text. This is done by simply maximizing the first part of the score: $\mathbf{z}_s = \arg\max_{\mathbf{z}_s} \mathbf{w}_s^T \mathbf{f}(G', G, \mathbf{t}, h, \mathbf{z}_s)$. Here, we only consider subsets of 1, 2 and 3 sentences as most questions can be answered by 3 sentences in the passage. The second step is formulated as an integer linear program by rewriting the scoring function. The ILP objective is:

$$\sum_{v' \in V'} \sum_{v \in V} z_{v',v} \mathbf{w}_v^T \mathbf{f}(v', v) + \sum_{e' \in E'} \sum_{e \in E} z_{e',e} \mathbf{w}_e^T \mathbf{f}(e', e)$$

Here, with some abuse of notation, $z_{v',v}$ and $z_{e',e}$ are binary integers such that $z_{v',v} = 1$ iff $\mathbf{z}$ maps v' onto v else $z_{v',v} = 0$. Similarly, $z_{e',e} = 1$ iff $\mathbf{z}$ maps e' onto e else $z_{e',e} = 0$. Additionally, we have the following constrains to our ILP:

- Each node $v' \in V'$ (or each edge $e' \in E'$) is mapped to exactly one node $v \in V$ (or one edge $e \in E$). Hence: $\sum_{v \in V} z_{v',v} = 1 \quad \forall v'$ and $\sum_{e \in E} z_{e',e} = 1 \quad \forall e'$
- If an edge $e' \in E'$ is mapped to an edge $e \in E$, then vertices $(v_{e'}^1, v_{e'}^2)$ that form the end points of e' must also be aligned to vertices (v_e^1, v_e^2) that form the end points of e. Here, we note that AMR parses also have inverse relations such as "arg0-of". Hence, we resolve this with a slight modification. If neither or both relations (corresponding to edges e' and e) are inverse relations (case 1), we enforce that $v_{e'}^1$ align with v_e^1 and $v_{e'}^2$ align with v_e^2. If exactly one of the relations is an inverse relation (case 2), we enforce that $v_{e'}^1$ align with v_e^2 and $v_{e'}^2$ align with v_e^1. Hence, we introduce the following constraints:

$$z_{e'e} \leq z_{v_{e'}^1, v_e^1} \textit{ and } z_{e'e} \leq z_{v_{e'}^2, v_e^2} \quad \forall e'.e \textit{ in case 1}$$

$$z_{e'e} \leq z_{v_{e'}^1, v_e^2} \textit{ and } z_{e'e} \leq z_{v_{e'}^2, v_e^1} \quad \forall e'.e \textit{ in case 2}$$

2.4 Features

Our feature function $\psi(\mathbf{t}, h, \mathbf{z})$ decomposes into three parts, each corresponding to a part of the latent structure.

The first part corresponds to relevant sentence selection. Here, we include features for matching local neighborhoods in the sentence subset and the hypothesis: features for matching bigrams, trigrams, dependencies, semantic roles, predicate-argument structure as well as the global syntactic structure: a graph kernel for matching AMR graphs of entire sentences (Srivastava and Hovy, 2013). Before computing the graph kernel, we reverse all inverse relation edges in the AMR graph. Note that if a sentence subset contains the answer to the question, it should intuitively be similar to the question as well as to the answer. Hence, we add features that are the element-wise product of features for the subset-question match and subset-answer match. In addition to features for the exact word/phrase match of the snippet and the hypothesis, we also add features using two paraphrase databases: ParaPara (Chan et al., 2011) and DIRT (Lin and Pantel, 2001). These databases contain paraphrase rules of the form string$_1$ → string$_2$. ParaPara rules were extracted through bilingual pivoting and DIRT rules were extracted using the distributional hypothesis. Whenever we have a substring in the text snippet that can be transformed into another using any of these two databases, we keep match features for the substring with a higher score (according to the current $\mathbf{w}$) and ignore the other substring. Finally, we also have features corresponding to the RST (Mann and Thompson, 1988) links to enable inference across sentences. RST tells us that sentences with discourse relations are related to each other and can help us answer certain kinds of questions (Jansen et al., 2014). For example, the "cause" relation between sentences in the text can often give cues that can help us answer "why" or "how" questions. Hence, we have additional features - conjunction of the rhetorical structure label from a RST parser and the question word as well.

The second part corresponds to node matches. Here, we have features for (a) Surface-form match (Edit-distance), and (b) Semantic word match (cosine similarity using SENNA word vectors (Collobert et al., 2011) and "Antonymy" 'Class-Inclusion' or 'Is-A' relations using Wordnet).

The third part corresponds to edge matches. Let the edges be $e = (v^1, r, v^2)$ and $e' = (v'^1, r', v'^2)$ for notational convenience. Here, we introduce two features based on the relations - indicator that the two relations are the same or inverse of each other, indicator that the two relations are in the same relation category – categories as described in Banarescu et al. (2013). Then, we introduce a number of features based on distributional representation of the node pairs. We compute three vertex vector compositions (sum, difference and product) of the nodes for each edge proposed in recent representation learning literature in NLP (Mitchell and Lapata, 2008; Mikolov et al., 2013) i.e. $v^1 \odot v^2$ and $v'^1 \odot v'^2$ for $\odot = \{+, -, \times\}$. Then, we compute the cosine similarities of the resulting compositions producing three features. Finally we introduce features based on the structured distributional semantic representation (Erk and Padó, 2008; Baroni and Lenci, 2010; Goyal et al., 2013) which takes the relations into account while performing the composition. Here, we use a large text corpora (in our experiments, the English Wikipedia) and construct a representation matrix $M^{(r)} \subset V \times V$ for every relation r (V is the vocabulary) where, the ij^{th} element $M^{(r)}_{ij}$ has the value log($1{+}x$) where x is the frequency for the i^{th} and j^{th} vocabulary items being in relation r in the corpora. This allows us to compose the node and relation representations and compare them. Here we compute the cosine similarity of the compositions $(v^1)^T M^{(r)}$ and $(v'^1)^T M^{(r')}$, the compositions $M^{(r)} v^2$ and $M^{(r')} v'^2$ and their repective sums $(v^1)^T M^{(r)} + M^{(r)} v^2$ and $(v'^1)^T M^{(r')} + M^{(r')} v'^2$ to get three more features.

2.5 Negation and Multi-task Learning

Next, we borrow two ideas from Sachan et al. (2015) namely, negation and multi-task learning, treating different question types in the machine comprehension setup as different tasks.

Handling negation is important for our model as facts align well with their negated versions. We use a simple heuristic. During training, if we detect negation (using a set of simple rules that test for presence of negation words ("not", "n't", etc.)), we flip the corresponding constraint, now requiring that the correct hypothesis to be ranked below all the incorrect ones. During test phase if we detect negation, we predict the answer corresponding to the hypothesis with the lowest score.

QA systems often include a question classification component that divides the questions into semantic categories based on the type of the question or answers expected. This allows the model to learn question type specific parameters when needed. We experiment with three task classifications proposed by Sachan et al. (2015). First is QClassification, which classifies the question, based on the question word (what, why, what, etc.). Next is the QAClassification scheme, which classifies questions into different semantic classes based on the possible semantic types of the answers sought. The third scheme, TaskClassification classifies the questions into one of 20 subtasks for Machine Comprehension proposed in Weston et al. (2015). We point the reader to Sachan et al. (2015) for details on the multi-task model.

3 Experiments

Datasets: We use MCTest-500 dataset (Richardson et al., 2013), a freely available set of 500 stories (300 train, 50 dev and 150 test) and associated questions to evaluate our model. Each story in MCTest has four multiple-choice questions, each with four answer choices. Each question has exactly one correct answer. Each question is also annotated as 'single' or 'multiple'. The questions annotated 'single' require just one sentence in the passage to answer them. For 'multiple' questions it should not be possible to find the answer to the question with just one sentence of the passage. In a sense, ‘multiple’ questions are harder than ‘single’ questions as they require more complex inference. We will present the results breakdown for ‘single’ or ‘multiple’ category questions as well.

Baselines: We compare our approach to the following baselines: (1-3) The first three baselines are taken from Richardson et al. (2013). *SW* and *SW+D* use a sliding window and match a bag of words constructed from the question and the candidate answer to the text. *RTE* uses textual entailment by selecting the hypothesis that has the highest likelihood of being entailed by the passage. (4) *LEX++*, taken from Smith et al. (2015) is another lexical matching method that takes into account multiple context windows, question types and coreference. (5) *JACANA* uses an off the shelf aligner and aligns the hypothesis statement with the passage. (6-7) *LSTM* and *QANTA*, taken from Sachan et al. (2015), use neural networks (LTSMs and Recursive NNs, respectively). (8) *ATTENTION*, taken from Yin et al. (2016), uses an attention-based convolutional neural network. (9) *DISCOURSE*, taken from Narasimhan and Barzilay (2015), proposes a discourse based model. (10-14) *LSSVM*, *LSSVM+Negation*, *LSSVM+Negation (MultiTask)*, taken from Sachan et al. (2015) are all discourse aware latent structural svm models. *LSSVM+Negation* accounts for negation. *LSSVM+Negation+MTL* further incoporates multi-task learning based on question types. Here, we have three variants of multitask learners based on the three question classification strategies. (15) Finally, *SYN+FRM+SEM*, taken from Wang et al. (2015) proposes a framework with features based on syntax, frame semantics, coreference and word embeddings.

			Single	**Multiple**	**All**
AMR		**Subgraph**	67.28	65.24	66.16
		Subgraph+Negation	69.48	66.46	67.83
	+MTL	**QClassification**	70.59	67.99	69.17
		QAClassification	71.32	68.29	69.67
		TaskClassification	**72.05**	**68.90**	**70.33**
Baselines		**SW**	54.56	54.04	54.28
		SW+D	62.99	58.00	60.26
		RTE	69.85	42.71	55.01
		LEX++	69.12	63.34	65.96
		JACANA Aligner	58.82	54.88	56.67
		LSTM	62.13	58.84	60.33
		QANTA	63.23	59.45	61.00
		ATTENTION	54.20	51.70	52.90
		DISCOURSE	68.38	59.90	63.75
		LSSVM	61.12	66.67	64.15
		LSSVM+Negation	63.24	66.15	64.83
	+MTL	**QClassification**	64.34	66.46	65.50
		QAClassification	66.18	67.37	66.83
		TaskClassification	67.65	67.99	67.83
		SYN+FRM+SEM	72.05	67.94	69.94

Table 1: Comparison of variations of our method against several baselines on the MCTest-500 dataset. The table shows accuracy on the test set of MCTest-500. All differences between the baselines (except *S*YN+FRM+SEM) and our approaches, and the improvements due to negation and multi-task learning are significant ($p < 0.05$) using the two-tailed paired T-test.

Results: We compare our AMR subgraph containment approach[2] where we consider our modifications for negation and multi-task learning as well in Table 1. We can observe that our models have a comparable performance to all the baselines including the neural network approaches and all previous approaches proposed for this task. Further, when we incorporate multi-task learning, our approach achieves the state of the art. Also, our approaches have a considerable improvement over the baselines for ‘multiple’ questions. This shows the benefit of our latent structure that allows us to combine evidence from multiple sentences. The negation heuristic helps significantly, especially for ‘single’ questions (majority of negation cases in the *MCTest* dataset are for the “single” questions). The multi-task method which performs a classification based on the subtasks for machine comprehension defined in Weston et al. (2015) does better than QAClassification that learns the question answer classification. QAClassification in turn performs better than QClassification that learns the question classification only.

These results, together, provide validation for our approach of subgraph matching over meaning representation graphs, and the incorporation of negation and multi-task learning.

4 Conclusion

We proposed a solution for reading comprehension tests using AMR. Our solution builds intermediate meaning representations for passage and question-answers. Then it poses the comprehension task as a subgraph matching task by learning latent alignments from one meaning representation to another. Our approach achieves competitive or better performance than other approaches proposed for this task. Incorporation of negation and multi-task learning leads to further improvements establishing it as the new state-of-the-art.

[2] We tune the SVM parameter C on the dev set. We use Stanford CoreNLP, HILDA parser (Feng and Hirst, 2014) and JAMR (Flanigan et al., 2014) for preprocessing.

References

[Banarescu et al.2013] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Baroni and Lenci2010] Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

[Chan et al.2011] Tsz Ping Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking bilingually extracted paraphrases using monolingual distributional similarity. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–42.

[Collobert et al.2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

[Cucerzan and Agichtein2005] S. Cucerzan and E. Agichtein. 2005. Factoid question answering over unstructured and structured content on the web. In *Proceedings of TREC 2005*.

[Davidson1969] Donald Davidson. 1969. *The individuation of events*. Springer.

[Erk and Padó2008] Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 897–906, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Feng and Hirst2014] Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521.

[Flanigan et al.2014] Jeffrey Flanigan, Sam Thomson, Jaime G. Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1426–1436.

[Goyal et al.2013] Kartik Goyal, Sujay Kumar Jauhar, Huiying Li, Mrinmaya Sachan, Shashank Srivastava, and Eduard H. Hovy. 2013. A structured distributional semantic model for event co-reference. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 467–473.

[Jansen et al.2014] Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 977–986.

[Lin and Pantel2001] Dekang Lin and Patrick Pantel. 2001. Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328.

[Mann and Thompson1988] William C Mann and Sandra A Thompson. 1988. {Rhetorical Structure Theory: Toward a functional theory of text organisation}. *Text*, 3(8):234–281.

[Mikolov et al.2013] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

[Mitchell and Lapata2008] Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 236–244.

[Narasimhan and Barzilay2015] Karthik Narasimhan and Regina Barzilay. 2015. Machine comprehension with discourse relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1253–1262.

[Parsons1990] Terence Parsons. 1990. *Events in the Semantics of English*, volume 5. In MIT Press.

[Richardson et al.2013] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

[Sachan et al.2015] Mrinmaya Sachan, Avinava Dubey, Eric P Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[Smith et al.2015] Ellery Smith, Nicola Greco, Matko Bosnjak, and Andreas Vlachos. 2015. A strong lexical matching method for the machine comprehension test. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1693–1698.

[Srivastava and Hovy2013] Shashank Srivastava and Dirk Hovy. 2013. A walk-based semantically enriched tree kernel over distributed word representations. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1411–1416.

[Wang et al.2015] Hai Wang, Mohit Bansal, Kevin Gimpel, and David A. McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 700–706.

[Weston et al.2015] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

[Yin et al.2016] Wenpeng Yin, Sebastian Ebert, and Hinrich Schtze. 2016. Attention-based convolutional neural network for machine comprehension. *arXiv preprint arXiv:1602.04341*.

[Yuille and Rangarajan2003] A. L. Yuille and Anand Rangarajan. 2003. The concave-convex procedure. *Neural Comput.*