AFRL-RI-RS-TR-2018-149

ROBUST DEEP SEMANTICS FOR LANGUAGE UNDERSTANDING

LELAND STANFORD UNIVERSITY

JUNE 2018

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2018-149 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ALEKSEY PANASYUK
Work Unit Manager

/ S /
JON S. JONES
Technical Advisor, Information Intelligence
Systems and Analysis Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUN 2018	FINAL TECHNICAL REPORT	OCT 2012 – DEC 2017

4. TITLE AND SUBTITLE

ROBUST DEEP SEMANTICS FOR LANGUAGE UNDERSTANDING

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
FA8750-13-2-0040

5c. PROGRAM ELEMENT NUMBER
62303E

6. AUTHOR(S)

Christopher Manning, Dan Jurafsky, Percy Liang

5d. PROJECT NUMBER
DEFT

5e. TASK NUMBER
12

5f. WORK UNIT NUMBER
11

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Leland Stanford Junior University
450 Serra Mall
Stanford, CA 94305

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2018-149

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Robust Deep Semantics for Language Understanding project, pursued under the Deep Exploration and Filtering of Text program, began with a research focus on five areas: deep learning, textual inferential relations, relation & event extraction by distant supervision, semantic parsing & ontology expansion, and coreference resolution. As time went by, the program focus converged towards emphasizing technologies for knowledge base population. The project successfully pioneered methods for deep learning for natural language understanding, effective knowledge base construction from text, natural logic methods for text understanding, improved mention coreference algorithms, and the further development of multilingual tools in CoreNLP.

15. SUBJECT TERMS
deep learning, neural networks, natural language understanding, coreference resolution, semantic parsing, natural logic, knowledge base population, relation extraction.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**ALEKSEY PANASYUK**
U	U	U	UU	38	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

1 Summary 1
2 Introduction 2
3 Methods, Assumptions, and Procedures 3
3.1 Fundamental Deep Learning Research 3
3.1.1 Overview 3
3.1.2 Details 3
3.2 Textual Inferential Relations 7
3.2.1 Natural Logic 7
3.2.2 Models of semantic textual similarity 9
3.3 Relation Extraction and Knowledge Base Population 9
3.3.1 Overview 9
3.3.2 Details 10
3.4 Semantic Parsing 14
3.4.1 Semantic Parsing for Freebase 14
3.4.2 Knowledge Extraction from Web Tables 15
3.4.3 Other work 16
3.5 Coreference Resolution 17
3.5.1 Overview 17
3.5.2 Details 17
3.6 Development of Stanford CoreNLP 19
3.6.1 Overview 19
3.6.2 Details 19
3.6.3 Integration with the BBN ADEPT framework 21
3.6.4 Impact 21
4 Results and Discussion 22
4.1 KBP Slot Filling and Cold Start Knowledge Base Population 22
4.1.1 Experience prior to this project 22
4.1.2 TAC KBP 2013 22
4.1.3 TAC KBP 2014 23
4.1.4 TAC KBP 2015 24
4.1.5 TAC KBP 2016 25

4.1.6 TAC KBP 2017 26
4.2 Other results 26
5 Conclusions 27
6 References 28
List of Symbols, Abbreviations, and Acronyms 32

List of Figures

Figure 1: Sentiment analysis by a Recursive Neural Tensor Network 6
Figure 2: Crowdsourcing interfaces for: (a) entity detection and linking; (b) relation extraction.13

List of Tables

Table 1: Stanford's KBP SF submissions for 2013..... 23
Table 2: Stanford's 2013 KBP results as compared to other teams that year 23
Table 3: Stanford's 2014 KBP results 24
Table 4: KBP 2015 system results on Hop All KB evaluation..... 25
Table 5: Macro-averaged LDC-MEAN KBP 2016 KB track Hop All scores..... 25
Table 6: Macro-averaged LDC-MEAN KBP 2017 KB track Hop All scores..... 26
Table 7: Coreference resolution CoNLL score on CoNLL 2012 test set 26

1 SUMMARY

The Robust Deep Semantics for Language Understanding project was pursued under the Deep Exploration and Filtering of Text program. It addressed the problem of how computer systems could effectively extract knowledge from text documents, from both formally written sources like newspaper articles and from informal sources such as web forums, and in multiple languages, with our work covering English, Spanish, and Chinese. Our major goal was to innovate on new methods for text understanding and knowledge extraction. The project did important work in developing the use of deep learning methods for natural language understanding and in developing new, improved algorithms for textual relation extraction and coreference resolution. While full text understanding is still far from a solved problem, the main goals of the project were achieved. Our group produced a variety of new and highly influential algorithms. During the early years of the project, our group produced much of the most cited work in using deep learning for natural language understanding, before use of these techniques disseminated more broadly. Our algorithms posted state-of-the-art results on a number of domains and tasks, and, partly through our making our algorithms broadly available open source in an integrated fashion through our CoreNLP software framework, they have had a considerable influence. The algorithms have seen considerable use, by many people in academia, government, the military, and industry.

2 INTRODUCTION

The Robust Deep Semantics for Language Understanding project pursued under the Deep Exploration and Filtering of Text (DEFT) program began with a focus on five areas: deep learning for natural language understanding, textual inferential relations, relation & event extraction by distant supervision, semantic parsing & ontology expansion, and coreference resolution. The overall goal was to develop a more effective means of understanding the meaning (semantics) of text in a robust and detailed way. Substantial fundamental research was done in all of these areas, as is discussed in following sections of this document. As time went by, the focus of the program converged in the direction of knowledge base population from text. That is, the question is how can we automatically convert a large collection of text documents to a knowledge base – or just database tables, if you prefer – which captures the entities that are mentioned in the text and the relations and events in which they are involved. The main foci of our research were relation extraction (or slot filling) for knowledge base population, coreference resolution, to detect mentions of the same entity, deep learning for NLP, and development of multilingual tools inside our CoreNLP software framework, so that text in multiple languages could be effectively ingested. Each year, our group participated in the NIST Knowledge Base Population evaluation, and this evaluation provides the key metric for our success and progress, but many individual algorithms were evaluated separately.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 FUNDAMENTAL DEEP LEARNING RESEARCH

3.1.1 Overview

While work on applying deep learning approaches to natural language processing (NLP) in the Stanford NLP Group had begun about two years prior to the start of the DEFT program, our continued work on this approach in DEFT led to many important early deep learning NLP papers. Although particular methods have evolved quickly, we acted as a standard bearer in the broader adoption of deep learning approaches to NLP. Most of the papers mentioned below already have several hundred or more citations on Google Scholar; two have over a thousand. Prominent deep learning NLP work done within this project includes:

- Developed a new recursive neural tensor network model and new corpus, the Stanford Sentiment Treebank, for sentiment classification using a compositional tree-structured analysis of sentences (Socher et al. 2013a). The dataset has been widely used in the subsequent years.
- Used morphological analysis of words to improve modeling the meaning of rare words, including development of a rare words similarity test set, still commonly used in evaluating word vectors (Luong et al. 2013).
- Developed a dependency tree-based recursive neural network and applied it to multimodal image and language similarity (Socher et al. 2014).
- Developed a compositional vector grammar parser for improved syntactic constituency parsing (Socher et al. 2013c).
- Developed an improved method for knowledge base completion with a neural tensor network. (Socher et al. 2013b).
- Developed a method of zero-shot learning in neural networks, improved by the use of cross-modal transfer (Socher et al. 2013d).
- Pioneered the use of neural network methods for dependency parsing, producing a fast and accurate neural dependency parser (Chen and Manning 2014). This was incorporated into CoreNLP, got many people motivated to explore use of neural networks for NLP, and it was the starting point for Google's work that led to their SyntaxNet and Parsey McParseFace parsers, now available in the Google Cloud Platform.
- Developed a Tree-LSTM model, which generalizes Long Short-Term Memory (LSTM) networks, a neural network architecture which has been used with great success recently for modeling sequential data, and applied it over trees to sentiment analysis (Tai et al. 2015).
- Built pioneering end-to-end neural reading comprehension and question answering systems (Chen et al. 2016).

During later parts of the project, building on our successful fundamental research on deep learning for NLP, we began actively exploiting use of deep learning methods for relation extraction and coreference resolution. That work is not described in this section but in sections 6.2 and 6.5 below.

3.1.2 Details

3.1.2.1 Natural language parsing

Natural language parsing has typically been done with small sets of discrete categories such as NP and VP, but this representation does not capture the full syntactic nor semantic richness of linguistic phrases and attempts to improve on this by lexicalizing phrases or splitting categories

only partly address the problem at the cost of huge feature spaces and sparseness. Instead, we introduced a Compositional Vector Grammar (CVG), which combines PCFGs with a syntactically untied recursive neural network that learns syntactico-semantic, compositional vector representations. The CVG improves the PCFG of the Stanford Parser by 3.8% to obtain an F1 score of 90.4%. It is fast to train and, implemented approximately as an efficient reranker, it is about 20% faster than the current Stanford factored parser (though slower than using the Stanford accurate unlexicalized PCFG parser alone). The CVG learns a soft notion of head words and improves performance on the types of ambiguities that require semantic information such as PP attachments. This line of work is further described in Socher et al. (2013c).

Subsequently, we also developed a fast and accurate dependency parser that is based on deep learning. Current dependency parsers come in two flavors – transition-based parsers, which sacrifice accuracy for speed (about 450 sentences/second with unlabeled attachment score of 89.4%) and graph-based parsers, which favor the reverse tradeoff (about 10 sentences/second with unlabeled attachment score of 90.7%). We developed a novel transition-based dependency parser using neural networks (NNs). The NN-based classifier learns and uses only a small number of dense features, rather than using millions of sparse indicator features. This is achieved by representing all words, part-of-speech tags and arc labels as dense vectors and modeling their interaction through a novel cubic activation function. As a result, our parser is extremely fast, while achieving about 2% improvement in unlabeled and labeled attachment scores on both English and Chinese datasets, as compared with other greedy transition-based parsers using indicator features. Our parser is able to parse more than 1000 sentences per second at 91.8% unlabeled attachment score for English. This NN dependency parser has been included in the Stanford CoreNLP system and is described further in Chen and Manning (2014). This work inspired a lot of people to try deep learning approaches and led to a lot of new work in further improving dependency parsing based on neural methods.

3.1.2.2 Word representations

We worked on extending neural word representations to understand morphologically complex words. Vector-space word representations have shown success in recent years at advancing state-of-the-art performance across a variety of NLP tasks. However, common to most existing work on word representations in latent spaces, words are regarded as independent entities without any explicit relationship among morphologically related words being modeled. As a result, rare and complex words are often poorly estimated, and all unknown words are represented in a rather crude way using only one or a few vectors. We have addressed this shortcoming by proposing a novel model that is capable of building representations for morphologically complex words from their morphemes. We combined recursive neural networks (RNNs), where each morpheme is a basic unit, with neural language models (NLMs) to consider contextual information in learning morphologically-aware word representations. Our learned models outperformed existing word representations by a good margin on word similarity tasks across many datasets, including a new dataset we introduced, focused on rare words to complement existing ones in an interesting way. This work is described in more detail in Luong et al. (2013).

Deep learning methods for learning vector space representations of words had already succeeded in capturing fine-grained semantic and syntactic regularities using vector arithmetic, but the origin of these regularities had remained opaque. We analyzed and made explicit the model properties needed for such regularities to emerge in word vectors. The result is a new global log-bilinear regression model that combines the advantages of the two major model families in the literature: global matrix factorization and local context window methods. Our

model efficiently leverages statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus. The model produces a vector space with meaningful substructure, as evidenced by its performance of 75% on a recent word analogy task. It also outperforms related models on similarity tasks and named entity recognition. This resulting GloVe model and word vectors using this technique that we distributed have been extremely widely used in NLP over the last five years. For at least four years after their publication, researchers found these to be the best word vectors for use in tasks like question answering and reading comprehension, although that may now be changing with the release of new word vectors from major U.S. technology corporations. This work is described in more detail in Pennington et al. (2014).

3.1.2.3 TreeRNNs and Sentiment Analysis

Our early research had a strong thread of trying to build tree-structured neural network models of language, in order to better capture the syntactic and semantic structure of sentences. Previous work of ours (Socher et al, 2011) on Tree Recursive Neural Networks (TreeRNNs) showed that these models can produce compositional feature vectors for accurately representing and classifying sentences or images. However, it had not yet been shown whether the RNN-induced representations could be useful for learning joint meaning representations for both modalities. We built a new DT-RNN model, which embeds sentences based on their syntactic dependency trees. Unlike previous RNN-based models that use constituency trees, DT-RNNs naturally focus on the action and agents in a sentence. They are better able to abstract from the details of word order and syntactic expression. DT-RNNs outperform other recursive and recurrent neural networks, kernelized canonical correlation analysis and a bag-of-words baseline on the tasks of finding an image that fits a sentence description and vice versa. They also give more similar representations to sentences that describe the same image. The DT-RNN model is more invariant and robust to surface changes in the sentences like word order. This work is described in more detail in Socher et al. (2014).

We developed a new Recursive Neural Tensor Network (RNTN) for modeling semantic composition. We demonstrated its effectiveness for predicting sentiment over sentence tree structures, showing it outperforms all of our previous recursive neural networks, our matrix-vector recursive neural network and several bag-of-words baselines. Figure 1 is one picture of the system's analysis of a contrastive sentence for negative/positive sentiment, with red (–) nodes showing sentiment negative words and phrases, and blue (+) nodes positive words and phrases. The example shows how the system correctly decides that the overall sentiment of the sentence is positive.

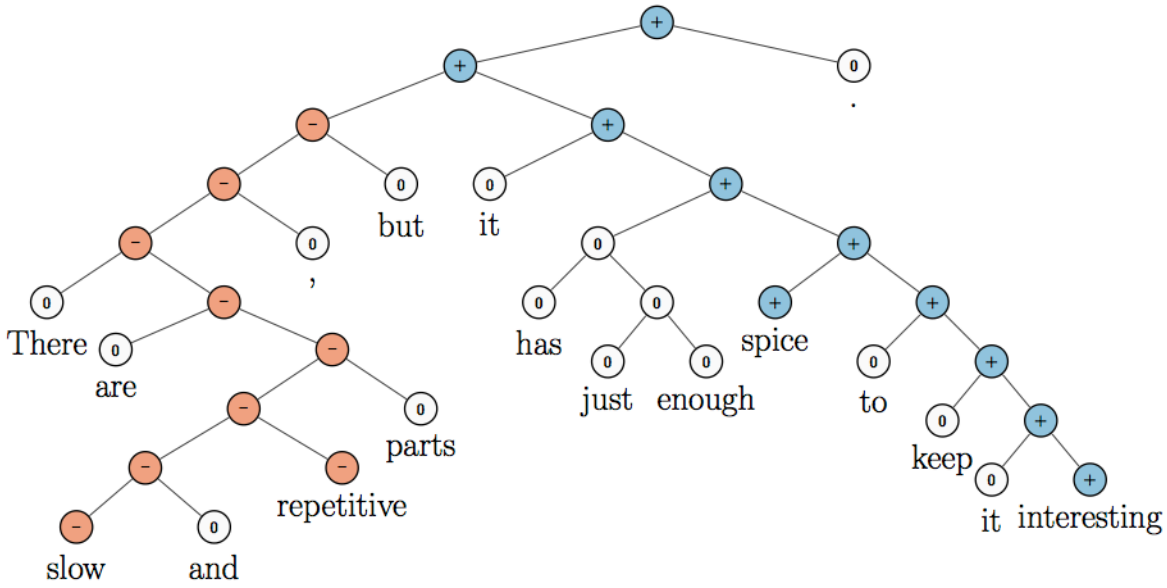

Figure 1: Sentiment analysis by a Recursive Neural Tensor Network

In order to understand compositionality in sentiment detection, we not only require new powerful compositional models but also a richer resource for supervision and evaluation. To address this, we have introduced a new annotated corpus called the Stanford Sentiment Treebank, which includes fine-grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences. This is the first corpus with sentiment-labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The Recursive Neural Tensor Network (RNTN) that can accurately predict compositional semantic effects is tested with this new corpus. When trained on the new treebank, this model out-performs all previous methods on several metrics. It pushes the state-of-the-art in single sentence positive/negative classification from 80% up to 85.4%. The accuracy of predicting fine-grained sentiment labels for all phrases reaches 80.7%, an improvement of 9.7% over bag of features baselines. Lastly, it is the only model that can accurately capture the effects of negation and its scope at various tree levels for both positive and negative phrases. This work appears in Socher et al. (2013a).

We later developed a new compositional tree-structured model for producing vector space representations of sentences. Our Tree-LSTM model generalizes Long Short-Term Memory (LSTM) networks, a neural network architecture which has been used with great success recently for modeling sequential data (e.g., Sutskever et al. 2014). Building on our previous work on tree-structured recursive neural networks, our Tree-LSTM model composes the vector representation corresponding to each node of a syntactic parse tree as a function of the vectors corresponding to the node's children. We demonstrated that Tree-LSTMs achieve new state-of-the-art results on two tasks: predicting the semantic similarity of sentence pairs and sentiment classification on the Stanford Sentiment Treebank. We then did additional experimental analysis to better understand how information is encoded and communicated within Tree-LSTM networks. This work is described in more detail in Tai et al. (2015).

3.1.2.4 Other topics

In another piece of work, we have defined a new structured prediction framework that allows structures to be represented in a distributional fashion. In recent years, distributional representations of inputs have led to performance gains in many applications by allowing statistical information to be shared across inputs. However, the predicted outputs (labels, or more generally, structures like trees) are still treated as discrete objects, even though outputs are also not discrete units of meaning. In our new formulation for structured prediction, we represent individual labels in a structure as real-valued vectors allowing semantically similar labels to share parameters. We extend this representation to larger structures by defining compositionality using tensor products and showed that our approach is a natural extension to standard structured prediction approaches. We proposed a learning objective for jointly learning the model parameters and the label vectors and defined an alternating minimization algorithm for learning. We applied our formulation to two tasks - multiclass document classification and English and Basque part-of-speech tagging (a sequence model) and outperform standard structured learning baselines. This work is described in Srikumar and Manning (2014).

In the latter part of the project, we started to expand our scope beyond single sentences and aim to develop end-to-end deep neural network systems for document-level understanding. Representing a whole document and extracting useful information remains a challenge, as 1) there are often thousands of words per document (compared to 20–40 words per sentence) and 2) it is important to handle the discourse structure between sentences as well as the syntactic structure within sentences. We developed a system for a reading comprehension task, that is, given a document and a cloze-style question, the goal is to infer the missing entity in the question based on an understanding of the document. The main idea is to apply sequential models (e.g., LSTMs) to encode all pieces of sentence-level (or even smaller text units) information within the document, and then to employ an attention mechanism (Bahdanau et al, 2015) to pick out the relevant snippets for retrieving the correct answer. On a recently released dataset from Google DeepMind (the DeepMind Daily Mail/CNN dataset), our system achieves 76% accuracy, with a single model (more with ensembling) exceeding the previous state-of-the-art results by more than 5% absolute. This work was an important early approach to end-to-end question answering models and was extended later for use on other datasets, at Facebook, the Toyota Technological Institute and elsewhere. The model is described in Chen et al. (2016).

3.2 TEXTUAL INFERENTIAL RELATIONS

3.2.1 Natural Logic

Our major initiative has been working on using Natural Logic (van Benthem 2014) for common sense reasoning with text. Natural Logic is logic whose syntax is natural language and inference is performed by executing truth-preserving transformations at the textual level. Although structured knowledge bases are powerful for querying information in restricted domains, much of the world's knowledge doesn't fit into these sorts of strictly structured KBs. In these cases, it's appealing to extract knowledge directly from text. For example, if a corpus contains the sentence "the cat ate a mouse" we should be able to infer that "no carnivores eat animals" is false. To accomplish this, we appeal to *natural logic* as a formalism that allows us to perform logical inference directly over the syntax of natural language. Stanford has been developing a large-scale natural logic reasoning engine which, given a web-scale corpus of facts, will infer whether a latent fact is true or false.

We attempted the task of *database completion*: given a database of true facts, we would like to predict whether an unseen fact should belong to the database. This is intuitively cast as an inference problem from a collection of candidate premises to the truth of the query. For example, we would like to infer that *no carnivores eat animals* is false given a database containing *the cat ate a mouse*. These inferences are difficult to capture in a principled way while maintaining high recall, particularly for open-domain text. Learned inference rules are difficult to generalize to arbitrary relations, and standard IR methods easily miss small but semantically important lexical differences. Furthermore, many methods require explicitly modeling either the database, the query, or both in a formal meaning representation (e.g., Freebase tuples).

Although projects like the Abstract Meaning Representation (Banarescu et al., 2013) have made headway in providing broad-coverage meaning representations, it remains appealing to use human language as the vessel for inference. Furthermore, Open Information Extraction (OpenIE) and similar projects have been very successful at collecting databases of natural language snippets from an ever-increasing corpus of un-structured text. These factors motivate our use of Natural Logic – a proof system built on the syntax of human language – to create a system for broad coverage database completion. In addition to being able to provide strictly valid derivations, our system is also able to produce derivations which are only *likely* valid, accompanied by an associated confidence. We have shown that our system is able to capture strict Natural Logic inferences on the FraCaS test suite and demonstrated its ability to infer previously unseen facts with 48% recall and 93% precision on a common sense reasoning evaluation, using data from the Ollie OpenIE system and a test set drawn from ConceptNet.

We proceeded to develop a system applying natural logic to knowledge base completion and question answering: NaturalLI. Rather than starting from a known schema-based knowledge base and inferring additional facts, NaturalLI operates over a plain-text knowledge base, and infers the truth of an arbitrary query based on the facts in this knowledge base. Natural logic itself is a formalism for inferring whether a sentence is entailed by another sentence based solely on the syntax of the sentence, rather than appealing to an explicit logical form. The NaturalLI question answering system was originally built to answer queries about common-sense facts in support of our development of probabilistic knowledge bases, but it was then extended to handle more complex real-world questions. To support this, a system was built to segment a long utterance into logically entailed, maximally concise clauses. This allows the system to digest articles with complex syntax into a set of atomic statements that can be searched over by NaturalLI. In addition, we developed a method for incorporating an evaluation function – akin to the evaluation functions in game-playing search – to assess whether a hypothesis is likely to be entailed by the knowledge base even if no proof derivation is found by the NaturalLI search. This allowed us to apply NaturalLI to the task of answering 4th grade multiple-choice science exams. We have shown that our system outperforms prior published work on the task, as well as strong IR baselines, achieving a score of up to 67% (74% on the training set). Future work will focus on applying this technique to a wider range of domains and datasets.

Moreover, we could apply this system to provide an open information extraction (OpenIE) system. Given a long utterance, the system splits the utterance into short, independent clauses and applies Natural Logic inference to find the maximally concise forms of those clauses. Then, a small set of surface and dependency patterns extract OpenIE triples from these shortened clauses. Evaluation of this system on the KBP 2013 evaluation shows that the system outperforms both the University of Washington's OpenIE-based submission for that year (by up

to 7 F1) and performs above the median and competitively with custom-trained relation extractors at 27 F1. This natural logic work is described in Angeli et al. (2014, 2015). Finally, we applied this technique to answering 4th grade science questions. To deal with the relatively syntactically complex sentences found in the source corpus of science texts, Stanford has developed a hybrid system that combines the benefit of strict logical reasoning with the broad-domain applicability of shallow statistical classifiers. In addition, we showed that natural logic inferences can be performed on dependency trees and can operate on partially distributed lattices other than the hypernym hierarchy – e.g., locations or relational entailment. The system achieves 67% accuracy on the test set of science questions, outperforming strong baselines and prior work.

3.2.2 Models of semantic textual similarity

Additionally, we worked on supporting the *SEM 2013 Shared Task on Semantic Textual Similarity. Semantic textual similarity (STS) is a graded measure of the degree of semantic similarity between two snippets of text. STS was first introduced as a shared task for SemEval 2012. Building on the success of the SemEval 2012 shared task, we co-organized STS as the shared task for this year's *SEM conference. For the primary STS evaluation task, the paired snippets correspond to two short statements, approximately one sentence in length. For 2013, we explored a new typed similarity task that assesses the similarity of typed fields (title, author, subject, description) in a structure database. The STS shared task was highly successful with 34 research sites submitting 89 systems.

The Stanford NLP group focused on refinement of the annotation instructions for the primary STS task. With the new instruction we were able achieve the following interannotator correlations across the different genres included in this year's competition:

- HDL: 85.0%
- FNWN: 69.9%
- OnWN: 87.2%
- SMT: 65.8%

As was done for the first STS evaluation, STS sentence pairs were annotated using crowdsourcing. We found that the genre and sources of data included in this year's evaluation made it harder to obtain good interannotator agreement. In order to achieve good correlations, we needed to refine the instructions from last year's annotation effort, as well as provide more training material for crowdsource annotators.

In 2014, we participated in the SemEval 2014 Task 1 of evaluating Semantic Relatedness of full sentences, using our deep learning models of semantic similarity. We got a Pearson correlation of 0.827 between the predicted similarity scores and the gold standard ratings for sentence-sentence relatedness. This put us in second place for this task (with the 3 top teams having very similar scores), and well above all other teams from the U.S.A.

3.3 Relation Extraction and Knowledge Base Population

3.3.1 Overview

During the project, we produced two completely new implementations of KBP systems, and in between did a lot of work in extending and improving our models. Initially we worked on extending and improving an approach to KBP slotfilling based on using distant supervision between text documents and structured information corresponding to Wikipedia content, that is, Freebase.

In later years, we built a different, pioneering approach to KBP based around the use of a model backed by a large, in-memory database and more use of supervised learning techniques. We worked to add neural relation extraction components to our slotfilling systems. As part of this work, we built a large, crowdsourced, hand-labeled, supervised relation extraction dataset, TACRED (Zhang et al. 2017).

We took part in the NIST TAC KBP evaluations each year 2013–2017, and our concerted work and new technologies for Slot Filling and Cold Start Knowledge Base Construction led to rapidly improving results: We went from being 6th place (about 6 F1 behind the leading team) in 2013 to being the leading or equal leading team (depending on the metric chosen) in all of the 2015, 2016, and 2017 evaluations. Furthermore, we expanded out our system from being English-only to being the only system that covered all of English, Chinese, and Spanish. We also explored how one could provide unbiased on-demand evaluation of KBP systems (Chaganty et al. 2017).

3.3.2 Details

3.3.2.1 TAC KBP Slotfilling: The first system

Initially, Stanford substantially reorganized the structure of and rebuilt its existing KBP Slotfilling system codebase, which was a distant-supervision approach to relation extraction, based on the multi-instance, multi-label (MIML) work of Surdeanu, et al. (2012). The aim of the makeover was to optimize for easier extensibility and greater modularity of individual components. This modularity greatly simplified running subtasks of the KBP Slotfilling challenge, such as the answer validation track; in addition, it allowed for monitoring the performance of subsystems and fine-grained testing, which both improves immediate performance and simplified future maintenance of the code. This work was necessary infrastructural work for enabling our renewed participation in the TAC KBP challenges, starting from TAC KBP 2013.

The new code was organized into five modules: information retrieval, datum processing, training, relation classification, and evaluation/slot filling. The first of these – information retrieval – provides the system with relevant documents and sentences when queried with an entity of interest (and an optional slot fill). This component encapsulates the entirety of the system's dependence on the IR system Lucene, and much of its dependence on external files in general. The datum processing component handles the aspects of sentence and datum processing which are not dependent on Lucene or other IR components. This includes adding auxiliary annotations and featurization. The training module implements the training procedure – primarily, reading the training data and constructing the training dataset. This dataset is then used to train one of the relation extractors in the relation classification package, such as Stanford's MIML-RE classifier (Surdeanu et al. 2012). The evaluation module handles the slot filling task at test time, including calling the relation extractor and applying a number of consistency and inference techniques on the resulting set of slot fills.

We implemented consistency and inference for slot filling fir the 2013 KBP Slot Filling challenge. Moreover, we also developed a novel active learning driven distant supervision system and applied it to the KBP task. We defined a new example selection criterion that uses several instantiations of the MIML relation extraction system to the most informative examples to present to the human annotators. This approach further boosts the KBP F1. Our redesigned system showed significant improvements over the previous evaluation of the Stanford KBP system, as is discussed further in section 6.

3.3.2.2 TAC KBP Slotfilling and Knowledge Base Population: The second system

Stanford developed a second new framework for KBP, optimized as a platform for quickly incorporating varied research contributions. The system makes use of a common initial linguistic annotation (basically, the output of Stanford CoreNLP), and a common pipeline for extracting relation mentions, entity linking, etc. Then, each relation extraction component can make use of this data in an easy to read format and output a list of relation triples based on the input. The framework then again reads these relation triples, and automatically processes them into a coherent knowledge base ready for evaluation. The framework is based around the Greenplum distributed database, much like DeepDive (Wu et al. 2015, Zhang et al. 2017), with the result that it is massively parallelizable, and can leverage the underlying database's consistency and speed. However, unlike DeepDive, no requirement is placed on either the inference or the relation extraction portions of the pipeline – the entire system can be interfaced with an arbitrary program reading from stdin and writing to stdout.

Starting in 2015, the Stanford NLP Group participated in the TAC KBP challenge using this new database-backed KBP system. We earned the top score among teams submitting Cold Start Slot Filling systems.

Stanford tackled the challenge of impoverished training data for KBP by appealing to a self-trained bootstrapping approach. In self-trained bootstrapping, a set of high-precision patterns was manually created (informed by the 2013 evaluation set) – a system in itself already competitive with many of the top teams. These patterns were then run on the TAC-KBP corpus to extract high-precision positive sentences for each relation. These sentences, along with randomly chosen negatives, are then used to train classifiers. This silver-standard data can then be used to train any number of classifiers. In practice, the extractions from these patterns were used as noisy training data for (1) a supervised classifier trained with logistic regression; and (2) a deep learning LSTM classifier. The inclusion of a small amount of supervised data, collected in Angeli et. al (2014), ensures that the model learns more than just the patterns provided. While both distant supervision and this bootstrapping approach provide noisy data, our performance on the TAC KBP task suggested that high-precision (if somewhat template-like) data is at least as if not more valuable than the more-varied-but-noisier distantly supervised data on which we and many other teams had previously relied.

In later years, we worked to progressively refine and improve this system. In 2016, we performed a thorough error analysis at each step of the pipeline that led to the following significant improvements: First, we improved our English fine-grained NER system by expanding its coverage on job title and GPE entities and enabled the capture of hierarchical GPE mentions (i.e., a GPE mention inside an organization mention). Second, we experimented with a new deep learning relation extractor that combines convolutional neural network and recurrent neural network (LSTM) and is trained on a large crowdsourced dataset. This new relation extractor significantly improves the recall of the entire English system. Third, we used a new SVM-based model ensembling technique to combine 5 pattern and rule-based relation extractors, a self-trained supervised extractor from our 2015 KBP system, and a new deep learning system as mentioned above. Finally, we recognized that one of the major source of losses in the KBP pipeline was entity detection. Shifting away from traditional CRF based NER systems, we built a neural NER system, based on a Bi-directional LSTM-CNNs-CRF model by Ma and Hovy (2015), which improved F1 scores to 87.43 F1 from our last year's score of 82.69. This also translated into gains on the TAC KBP 2017 task with our system gaining 2–3 F1 points across different metrics. The improved systems were used in our final year KBP systems.

3.3.2.3 Neural network relation extraction systems and TACRED

Deep learning methods had been under-investigated for the TAC KBP slot filling tasks, mainly due to two reasons: (1) existing deep learning models are insufficiently tailored for the relation extraction problem; and (2) a large-scale dataset that is better customized for the TAC KBP slot filling tasks did not exist. In 2016, we began jointly tackling these two problems. First, we designed a neural architecture that makes use of a novel position-aware attention mechanism. This model is better tailored for the relation extraction task as it learns representations of the relation by jointly modeling the semantic information and the entity positions in the sentence being classified. Second, in order to power the new neural model, we collected a new large-scale dataset named the TAC Relation Extraction Dataset (TACRED), mainly via crowdsourcing. This dataset was created by making use of the TAC KBP 2009–2014 evaluation corpus and reusing the 41 original TAC KBP relation types. The resulting dataset contains 106,264 examples – an order of magnitude larger than the largest existing relation extraction dataset.

The combination of the new TACRED dataset and the novel neural architecture leads to state-of-the-art relation extraction and slot filling performance. In an evaluation on the relation extraction task using TACRED, our neural model has outperformed the best previous neural architecture by 3.5% in F1 score. When our neural model was explored in concert with a pattern-based system on the TAC KBP 2015 Cold Start Slot Filling evaluation task, the system achieves an F1 score of 26.7%, which exceeds the 2015 slot filling winning system by 4.5%. This system was used as part of an ensemble in our final year TAC KBP system. The ensembling of this model with other extractors gave the highest recall and was the best-performing model on the TAC KBP 2017 evaluation for the English slot-filling task.

3.3.2.4 TAC KBP Slotfilling: Building a multilingual system

The TAC KBP 2016 challenge introduced two new multilingual tracks: apart from English, there was now Chinese and Spanish, as well as a new cross-lingual track. We worked that year on the development of an entirely new Chinese KBP system. For our 2016 Chinese system, our contribution can be summarized as follows: First, we developed a new Chinese fine-grained NER system that consists of a retrained statistical tagger, a rule-based numeric NER tagger and a gazetteer-based tagger. This new system pushed the number of supported NER labels to 22 (from the original 5 labels). Second, we employed a new gazetteer-based Chinese entity linking system. Third, we developed five entirely new Chinese relation extractors, namely two pattern-based extractors, a distantly supervised extractor, and two rule-based extractors. We used a model combination of these five extractors to form our final submissions. Our final slot filling submissions outperformed all other teams in almost all of the English evaluation measures and led on some Chinese evaluation measures in the official evaluation.

In 2017, we have made several improvements to the Chinese KBP system. First, the Chinese system now has better processing ability for XML tags in discussion forums. We made use of this improvement to incorporate more information, such as post authorship, into the annotation of discussion forum text. Second, we made improvement of the Chinese NER systems, by jointly improving our gazetteers and augmenting our training data, and to the pattern-based and statistical Chinese relation extractors. We added new surface pattern-based rules to our rule-based extractor and a new specialized extractor for extracting location of headquarters from the name of an organization.

In 2017, we also developed a new KBP system for Spanish. The system combines statistical, rule-based, and gazetteer-based NER taggers to perform fine-grained NER for Spanish. We built gazetteers for a new fine grained NER, which improves relation extraction for certain relation

types. We also added a new HeidelTime date annotator, to extract dates and hence date relations. It then does entity linking with a gazetteer-based system. We retrained our CRF based NER systems for Spanish using more annotated data, in particular using in-domain data for discussion forums which improved performance on Spanish from 59.8 F1 to 73.2 F1. Our system for Spanish relation extraction was rule-based. We built two pattern-based relation extractors: one based on token-level regular expressions over sequences and one using dependency-tree regular expressions. We constructed more than 2,400 surface patterns and 500 dependency-based patterns. However, a lack of good dependency parses was preventing the latter extractor from performing well. We were able to incorporate our separate work on a parser for CoNLL 2017 "Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies" for Spanish, which was the best performing parser and POS tagger in that evaluation (Dozat et al. 2017). This deep neural network parser had a graph-based architecture and a bi-affine attention mechanism which greatly improved the parses and hence drastically improved the F1 score of our relation extractor. There is still ample room to further extend this system to handle dropped pronouns and coreference as well as explore using machine learning relation extractors.

3.3.2.5 Removing bias in TAC KBP system evaluation

A key problem in improving relation extraction system performance that we identified was that evaluation using previous years' assessments with a closed-world assumption leads to a significant bias against novel systems. This is particularly disadvantageous for machine learning based approaches as the systems are penalized for predicting novel but presumably correct relations. While the TAC KBP challenge doesn't suffer from this bias, it is not amenable to a short development cycle. As a solution, we proposed a new evaluation methodology, on-demand evaluation, which avoids pooling bias by querying crowd workers, but to minimize cost, does it selectively, leveraging previous systems' predictions when possible. We then compute the new system's score based on the predictions of past systems using importance weighting. We implemented our framework and made a publicly available evaluation service where researchers can evaluate their own KBP systems. We piloted this service by evaluating three distinct systems on the 2016 TAC KBP corpus for about $300 a system (a fraction of the cost of official NIST evaluation). The Mechanical Turk crowdsourcing interface for this system is shown in Figure 2. This system is further described in Chaganty et al. (2017).

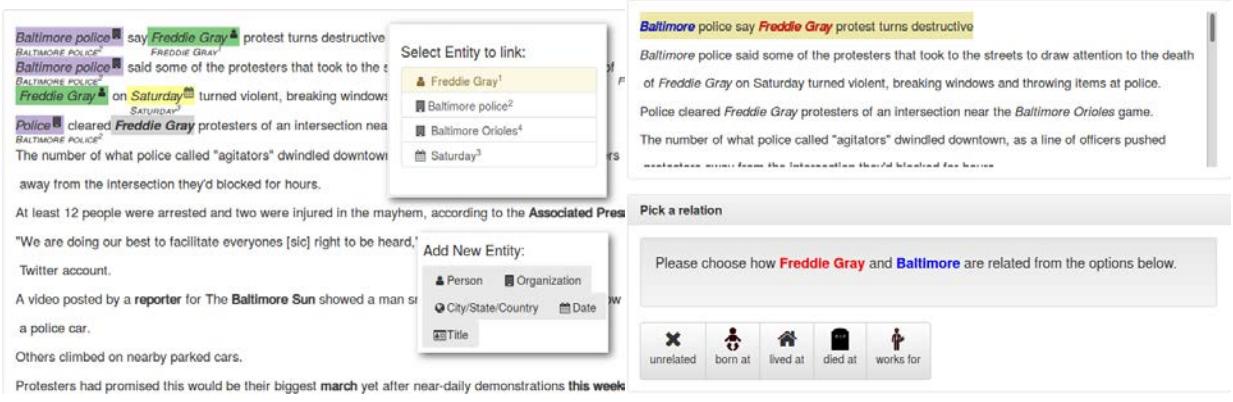

Figure 2: Crowdsourcing interfaces for: (a) entity detection and linking; (b) relation extraction

3.3.2.6 Other work

In 2017, together with colleagues from Rensselaer Polytechnic Institute (RPI), the University of Illinois at Urbana-Champaign (UIUC), Cornell University, Columbia University, Johns Hopkins

University, and the University of Pennsylvania, we built a joint system, Tinkerbell, which covered a complete knowledge base construction from textual input project, including, relations, events, beliefs, sentiment, and entity linking. Our contribution was to do the basic linguistic analysis (via CoreNLP) and the relation extraction for all three languages (Chinese, Spanish, and English). This effort allowed us to experiment with the work on entity linking from RPI and UIUC.

Early in the project, we applied neural networks to the task of completing a knowledge base. Knowledge bases provide applications with the benefit of easily accessible, systematic relational knowledge but often suffer in practice from their incompleteness and lack of knowledge of new entities and relations. Much work has focused on building or extending them by finding patterns in large unannotated text corpora. In contrast, we did research aiming to complete a knowledge base by predicting additional true relationships between entities, based on generalizations that can be discerned in the given knowledge base. We introduced a neural tensor network (NTN) model which predicts new relationship entries that can be added to the database. This model can be improved by initializing entity representations with word vectors learned in an unsupervised fashion from text, and when doing this, existing relations can even be queried for entities that were not present in the database. Our model generalized well and outperformed several existing models for this problem and can classify unseen relationships in WordNet with an accuracy of 75.8%. This work is described in Socher et al. (2013b).

Additionally, Stanford worked on an AMR aligner and parser, exploring a number of techniques for both. We have shown that an aligner similar to the IBM models for machine translation—adapted to a sentence-to-graph scenario—performs well at the task; with the addition of domain-specific rules we have created an aligner which they believe is state-of-the-art. We worked on creating a full end-to-end AMR parser. Approaches ranged from a CCG-like CKY parser, a greedy shift-reduce parser, and a hierarchical recursive HMM procedure. The best performing system is a translation from a number of NLP components to their associated AMR syntax, including the extended semantic roles of Srikumar and Roth (2011), the output of CoreNLP, among other components. We had hoped that this work could improve relation and event extraction, but this wasn't realized within the bounds of this project. This work is further described in Werling et al. (2015).

3.4 SEMANTIC PARSING

3.4.1 Semantic Parsing for Freebase

Semantic parsing focuses on mapping natural language utterances into logical forms that can be executed against a knowledge-base. Traditional approaches for semantic parsing have two limitations (a) they require annotated logical forms as supervision (b) they operate in limited domains where the number of relations in the ontology is small. We have addressed these limitations and developed a semantic parser that is trained over question-answer pairs and scales up to thousands of both atomic and composite relations in a large knowledge-base with a complex ontology (Freebase).

A major challenge in scalable semantic parsing is covering the large ontology (the large number of knowledge base relations). We handle this challenge using two complementary strategies: (a) we constructed a lexicon that maps natural language phrases to atomic and composite Freebase relations by aligning a large text corpus to Freebase; (b) We introduced a novel "bridging" operation that suggests relations that are compatible with entities and other logical predicates present in the context of a particular query. We developed a semantic parser

that scales up to Freebase, which is a large knowledge base consisting of more than 40 million entries and 600 million facts. Furthermore, instead of relying on annotated logical forms, which is especially expensive to obtain at large scale, we learn from question-answer pairs. We developed a framework to reliably determine the semantically correct logical form from partial supervision. One recent trend for training a semantic parser is to use the value resulting from executing the logical form, rather than the logical form itself, for supervision. This training signal is insufficient for more complex sentences, which can yield many semantically incorrect parses that execute to the correct answer. Our framework efficiently searches for all candidates' logical forms that execute to the correct answer, and then use either an unsupervised alignment approach or a small amount of human annotation to pick the semantically correct one. The final system is able to understand the meaning and compositionality of complex logical operations such as "the most", "average", "difference", and "same as". The new semantic parser uses two novel operations to tackle this problem of mapping natural language phrases to predicates in the Knowledge-base (KB). The first operation is alignment, in which we use a large snapshot of the web and OpenIE tools to align text phrases against the KB. The second operation is bridging, in which we suggest KB predicates that are expressed implicitly in the question using other KB predicates that are expressed more explicitly. For example, given the question "Who did Tom Cruise Marry in 2006?" we can combine alignment and bridging to construct the complex logical form $\lambda x.\exists y.marriage(x,y) \wedge spouse(y,TomCruise) \wedge startDate(y,2006)$, which evaluates on Freebase to "Katie Holmes".

We evaluated our parser on a recently released data set (Cai and Yates, 2013) that covers more than 600 Freebase relations. We found our parser outperformed a state-of-the-art parser (62.7% to 59%), despite the fact that its training signal is weaker (question-answer pairs rather than logical forms). We then collected a new and realistic data set of questions by performing breadth-first search using the Google Suggest API, and generated answers with Amazon Mechanical Turk. On this challenging data set we have achieved 35.7%, an 8.3% improvement over a natural baseline. This work was published as Berant et al. (2013) and was made available open source at https://nlp.stanford.edu/software/sempre/.

A central challenge in semantic parsing is handling the myriad ways in which knowledge base predicates can be expressed. Traditionally, semantic parsers are trained primarily from text paired with knowledge base information. Subsequently, we developed a new semantic parsing approach that exploits the much larger amounts of raw text not tied to any knowledge base. Given an input utterance, we first use a simple method to deterministically generate a set of candidate logical forms with a canonical realization in natural language for each. Then, we use a paraphrase model to choose the realization that best paraphrases the input and output the corresponding logical form. We defined two simple paraphrase models, an *association* model and a *vector space* model, and trained them jointly from question-answer pairs. Our system PARASEMPRE improved state-of-the-art accuracies on two recently released question-answering datasets. We believe that our approach opens a window of opportunity for learning semantic parsers from raw text not necessarily related to the target KB.

3.4.2 Knowledge Extraction from Web Tables

Semantic parsing also requires a KB with a fine-grained ontology. We worked on building such a KB from semi-structured web pages. Existing information extraction systems rely on seed examples or redundancy across multiple web pages. We considered a new zero-shot learning task of extracting entities specified by a natural language query (in place of seeds) given only a single web page. Our approach defines a log-linear model over latent extraction predicates, which

select lists of entities from the web page. We tackled the challenge of defining features on widely varying candidate entity lists by abstracting list elements and using aggregate statistics to define features. Finally, we created a new dataset of diverse queries and web pages and showed that our system achieves significantly better accuracy than a natural baseline.

We developed a novel parsing algorithm that learns a strategic policy for exploring this space, and substantially reduces the number of logical forms considered by the parser. The algorithm controls parsing by choosing the highest-scoring parsing action from an agenda, and learns an appropriate scoring function from data, using a reinforcement learning algorithm. We demonstrated that our parser is 8 times faster than prior work, while obtaining state-of-the-art accuracy.

Our work aims to increase both the *breadth* of the knowledge source and the *depth* of logical expressiveness by training a system to analyze an HTML table and answer a complex question based on the table. The absence of a fixed data schema in semi-structured Web tables enables question answering on knowledge from a much broader domain. Our algorithm allows the table to influence the construction of parse trees, making it generalize well to even previously unseen tables. In addition, our parsing algorithm uses highly recursive deduction rules to construct parse trees with expressive logical predicates, making it able to handle a wider scope of logical operations and deeper linguistic compositionality. In addition to the algorithm, we also released a large dataset of this task to encourage research on semantic understanding of semi-structured data.

3.4.3 Other work

Another challenge of semantic parsing is learning the semantic lexicons of new domains. To approach it, we developed a new framework for quickly building a semantic parser in a new domain without initial examples from that domain. In essence, given a pair of an in-domain seed lexicon and the syntax of the target logical form, the framework generates new logical forms along with a canonical natural language gloss and asks people to paraphrase the gloss. From the paraphrased sentences, the system can not only learn new words for concepts in the new domain (e.g., "attend *X*" = education is *X*), but also learn sub-lexical compositionality where a short phrase describes a complex concept (e.g., "mother of *X*" = parent of *X* who is female). Within a few hours of such human "training", the framework was able to build semantic parsers for seven different domains (Wang et al. 2015).

We also examined mapping descriptions of scenes to 3D geometric representations. Prior work on the text to 3D scene generation task has used manually specified object categories and language that identifies them. We introduced a dataset of 3D scenes annotated with natural language descriptions and learn from this data how to ground textual descriptions to physical objects. Our method successfully grounds a variety of lexical terms to concrete referents, and we show quantitatively that our method improves 3D scene generation over previous work using purely rule-based methods. We evaluated the fidelity and plausibility of 3D scenes generated with our grounding approach through human judgments. This work is described more fully in Chang et al. (2014, 2015).

Finally, we experimented with training a sequence-to-sequence neural semantic parsing model using inferred logical forms as supervision. Our analysis reveals that the model effectively learns general logical form structures (with 92% accuracy on artificial data) but has some challenges at identifying lexical items and distinguishing contextual words in the utterance from content words. In future work, we intend to undertake an additional project that addresses these challenges directly.

3.5 COREFERENCE RESOLUTION

3.5.1 Overview

We initially explored methods to improve the rule-based or "sieve" coreference system that we had built in 2011 (winning the CoNLL 2011 Shared Task on Coreference over OntoNotes data). We added a classifier to detect singleton mentions and methods for better modeling of full nominal coreference emphasizing the difference between what is "similar" (e.g., "Facebook" and "Google") vs. what can be coreferent (e.g., "Google" and "the search giant") and started to build a hybrid rule-based and machine learning system.

Subsequently, we began exploring fully learned approaches to coreference resolution. We first built a statistical machine learning coreference system, which was based around entity models (Clark and Manning 2015). Subsequently we built two versions of neural coreference systems, one again based on entities but now using neural similarity functions (Clark and Manning 2016a) and the other just a mention-pair classifier trained via imitation learning (Clark and Manning 2016b). All three of these systems were the best-performing English coreference system at the time of their publication.

We also extended both our sieve coreference system and our neural coreference systems to Chinese and did some limited exploration of coreference for Spanish.

3.5.2 Details

3.5.2.1 Extending our rule-based and hybrid "sieve" coreference systems

We initially explored methods for improving our existing (rule-based) coreference resolution, which had won the CoNLL 2011 Shared Tasking on coreference over OntoNotes data (Lee et al. 2011). One is a classifier that attempts to detect via discourse features singleton mentions, which should not be coreferent with anything (Recasens et al. 2013). This work got the NAACL 2013 best short paper award. The initial work was prior to the DEFT project, but the DEFT project allowed us to integrate this work into the Stanford CoreNLP deterministic coreference resolution system, giving about a 0.6 F1 improvement on CoNLL 2011 F1.

We then developed a new hybrid statistical sieve system for English. The new architecture replaces rule-based sieves with statistical ones that are based on random forests to model feature interactions. In addition, we have also incorporated richer linguistic information (such as discourse), new datasets and higher precision context information to address linking common nouns. We worked to make this new hybrid statistical sieve system for English faster and lighter. One of the biggest obstacles to fast coreference on documents is that until now our coreference systems required constituency parsing, and our constituency parser was quite slow. The improved system relies solely on a dependency parser rather than a constituent parser. By swapping parsers, we incur a very small (about 1%) loss in performance, but the system is much faster, since we can use it with our neural dependency parser, which is orders of magnitude faster than constituency parsers. We also removed some expensive features that weren't that important in improving performance, with the result that the model is smaller than before. Using this, we successfully incorporated the coreference resolver into our KBP system, which increased the system's recall by allowing it to process pronominal mentions. Second, we worked on porting our sieve-based coreference system to Chinese. We released in CoreNLP a deterministic sieve-based Chinese coreference resolution system that demonstrates comparable performance with the then state-of-the-art systems on the CoNLL 2012 Chinese coreference resolution task.

3.5.2.2 New statistical and neural network approaches to coreference

Beginning in late 2014, we began exploring new machine learning approaches to coreference resolution, initially an entity-centric probabilistic model and then a series of neural network-based coreference systems.

We first developed a coreference system that learns an effective policy for incrementally building up coreference clusters. The system operates by greedily merging clusters of mentions it predicts are likely to be coreferent. Training a system to do this is challenging because (1) it requires defining features between clusters of mentions instead of pairs and (2) the number of possible cluster merges is large. We address these challenges by decomposing the learning problem into two steps. First, we train classifiers that predict whether or not two mentions are coreferent. The scores produced by these models are then used to define powerful features between clusters of mentions and prune which candidate cluster merges are considered. Using this feature set and constrained search space, our system is able to learn when two clusters of mentions should be merged with an imitation learning algorithm. Our system achieves a CoNLL F1 score of 63.0 on the OntoNotes 5.0 corpus, the highest reported score to date. This system was incorporated into CoreNLP as a fast statistical coreference system and is described further in Clark and Manning (2015).

Thereafter, combining neural networks and coreference, we developed several neural-network-based coreference systems that learn to produce effective vector representations of mention pairs. These systems do not rely on the complex highly engineered features commonly used in other coreference systems, which can become unwieldy and may generalize poorly to new data. We developed a neural-network based coreference system that uses a much smaller set of features, instead relying on distributed word representations to inform the model.

The central component of the first new system is a neural network that produces high-dimensional distributed representations of pairs of coreference clusters. Using these representations, our system learns when combining a pair of clusters is desirable. This allows coreference resolution to be performed with agglomerative clustering: initially, each mention is placed in its own singleton coreference cluster, then a pair of clusters is merged each step. We found applying this clustering algorithm to significantly improve accuracy over the commonly used mention-pair approach to coreference resolution. Training a clustering coreference system is challenging because the coreference decisions facing a model depend on previous decisions it has already made. We addressed this by applying a learning-to-search algorithm that teaches the model which local decisions (cluster merges) will eventually lead to a high-scoring final coreference partition. The resulting system substantially improved upon the current state-of-the-art over the OntoNotes dataset and particularly excels at hard coreference resolution problems that require knowledge about semantic similarity to solve (e.g., “the country” and “the nation”). This work is described in Clark and Manning (2016a).

Coreference resolution systems typically operate by making sequences of local decisions (e.g., adding a coreference link between two mentions). However, the goal of coreference is to have a desirable global structure (a complete set coreference clusters). Due to this difficulty, coreference systems are usually trained with loss functions that heuristically define the goodness of a particular coreference decision. These losses contain hyperparameters that are carefully selected to ensure the model performs well according to coreference evaluation metrics. Relying on hyperparameters complicates training, especially across different languages and datasets where systems may work best with different settings of the hyperparameters. Instead, we have addressed this challenge by training coreference models and reinforcement learning algorithms.

This directly optimizes models for coreference evaluation metrics, obviating the need for slow hyperparameter search. The approach also yields significant gains in accuracy. This work is described in more detail in Clark and Manning (2016b). This fully-learned system also gave us an easy opportunity to build systems for other languages, and so we also built a Chinese coreference system using OntoNotes data, which also achieved a new state-of-the-art performance. The new coreference systems were incorporated into our Java code for release in CoreNLP 3.7.0.

Finally, we have improved the performance of our English coreference resolution system by changing the mention detection system. In particular, we eliminated various heuristic rules for filtering candidate mentions, which we had inherited from the earlier sieve-based coreference system, instead relying on the model to identify singleton mentions. This change actually improved the system's CoNLL F1 score by 1.3 points on the English CoNLL 2012 data. Both developments will be incorporated into our TAC 2017 CS KBP System and the next release of Stanford CoreNLP.

We also developed a new coreference resolution system for Spanish trained on the AnCora dataset. The architecture and training of the system is the same as the neural-network based system we previously built for English and Chinese. The system has the ability to detect and add coreference links to dropped pronouns, which occur frequently in Spanish. However, despite the new coreference resolution system for Spanish being able to detect and add coreference links to extract dropped pronouns, it did not perform well given the parses that were available during initial stages of development. To fix the issue, we also developed a rule-based system for coreference for connecting similar named entities via fuzzy string matching. This system gave better results than the previously trained coreference system, given the bad quality of parses and was incorporated into our KBP pipeline.

3.6 DEVELOPMENT OF STANFORD CORENLP

3.6.1 Overview

Both for our own work and in general support of the DEFT program, we made many additions and improvements to our Stanford CoreNLP processing suite (Manning et al. 2014), including:

- We added direct dependency parsing via a neural dependency parser.
- We added POS tagging, NER, and constituency and dependency parsing for Spanish.
- We added improved NER and coreference models for Chinese.
- We added natural logic and open information extraction annotators (Angeli et al. 2015).
- We added a KBP relation extraction annotator for all of English, Chinese, and Spanish.
- We added new statistical and neural coreference systems.
- We added a web services API for CoreNLP.
- We added support for calling annotators in an annotation pipeline that are implemented as a web service.

3.6.2 Details

Throughout the project we made many extensions and improvements to our Stanford CoreNLP software toolkit for natural language processing, to improve analysis components, to add new analysis components, and to extend coverage of components to more languages.

The 2012-11-12 release added some training data derived from Wikipedia to the English NER models, improved and sped up the Stanford Dependencies code in the Stanford Parser, upgraded SUTime, and included various other bug fixes.

The 2013-04-04 version 1.3.5 release included the singleton detection research to improve coreference resolution, which was mentioned above, speed improvements including further multithreading support, some Stanford Dependencies improvements, and included various other bug fixes.

The 2013-06-20 version 3.2.0 release included a new and more accurate parser model. Additionally, it also incorporated a faster tagger and other bug fixes.

The 2013-11-12 version 3.3.0 release added a state-of-the-art deep learning–based sentiment analysis model. The releases also include improvements to English and Chinese dependency parsers, improvements to time entity recognition and bug fixes. The 2014-01-04 version 3.3.1 release fixed a few bugs.

The 2014-06-16 version 3.4.0 release added a much faster shift-reduce constituency parser.

The 2014-08-27 version 3.4.1 release added support for processing Spanish (part-of-speech tagging, named entity recognition, and constituency parsing. This was also the final release that supported Java 6 and Java 7.

The Stanford CoreNLP version 3.5.0 (2014-10-31), 3.5.1 (2015-01-29), and 3.5.2 (2015-04-20) releases made major enhancements in the areas of parsing and coreference. Version 3.5.0 was the first release of our new, much, much faster neural network-based dependency parser. Version 3.5.1 added a natural logic annotator and one for picking out quoted material. Starting in version 3.5.2, the parser now outputs grammatical relations in the new Universal Dependencies representation (http://universaldependencies.github.io/docs/) by default (although the traditional Stanford Dependencies are also available, by specifying a property). Another major new feature in version 3.5.2 is the ability to perform co-reference resolution for Chinese. In addition to these changes, version 3.5.2 also includes improved new models and refinements for named entity recognition.

The 2015-12-09 version 3.6.0 release includes the new more accurate statistical coreference system, including a faster coreference model that works on dependency parses, added our new OpenIE annotator, and introduced a StanfordCoreNLPServer web service, which has transformed the accessibility of CoreNLP to programmers working in non-JVM languages.

The 2016-10-31 version 3.7.0 release made substantial improvements. It includes neural-network based coreference systems for English and Chinese that are significantly more accurate than the previous coreference resolvers. We added a whole new set of higher accuracy dependency parser models that produce Universal Dependencies for English, Chinese, Spanish, Arabic, French, and German. There are also higher quality models for POS, NER, and Constituency Parsing in Spanish trained with Latin American Spanish materials, a new, improved German NER model, and improvements to Chinese NER including capturing quantifiable entities. The Stanford CoreNLP Server had bug fixes and improvements that allow it to work with non-English languages, match Tregex patterns (Levy and Andrew 200 on constituency trees and visualize constituency trees. There were also a variety of miscellaneous enhancements and bug fixes including improved code organization for coreference, improvements to the QuoteAnnotator, and the addition of sentiment extraction to the Simple API.

The 2017-06-09 version 3.8.0 release included several enhancements. There was a new web services annotator class to help facilitate adding non-Stanford NLP algorithms to a Stanford CoreNLP pipeline. A non-Stanford algorithm can be wrapped in a server, and this new web services annotator will start the server, submit requests and integrate the responses into an annotation. This will be especially helpful for using Python based tools with Stanford CoreNLP,

for now all one has to do is wrap their Python tool in a server and they can smoothly integrate it into a Java Stanford CoreNLP pipeline. For instance, we use this new type of annotator to integrate RPI Entity Linking information with our standard NLP tools. Also, a new annotator has been added for attributing quotes to entities in text, the discussion forum handling has been improved, and there will be new models for processing Spanish and French text. The core processing now extends to handling emoji (!), rather than only handling the Unicode Basic Multilingual Plane. We have also fixed bugs and various issues with our system based on user feedback. We added new UD POS models for French and Spanish. Lastly, the release will include improvements to our coreference system.

Finally, slightly after the official end of the DEFT program, we released CoreNLP version 3.9.0 (2018-01-31) and version 3.9.1 (2018-02-27) which incorporated work that we did during the final months of the DEFT program, as well as a bunch of unrelated work on better handling other languages. The release included the Spanish KBP model, better NER, and new dependency parse model. It improved French tokenization, UD POS tagging and parsing; provided better German, Chinese NER; added an Arabic SR parser model; added a wrapper API for certain data types, quote attribution improvements; provided easier use of coreference information; and included miscellaneous bug fixes and minor enhancements.

3.6.3 Integration with the BBN ADEPT framework

Throughout the DEFT program, we also did work in integrating our software with the BBN Adept framework. In the early years of DEFT we made separate releases of our KBP slotfilling software, incorporated into the BBN Adept system. In later years of the program, we added KBP relation extraction as a capability of CoreNLP, which broadened the availability and usability of this system. Nevertheless, we still did considerable work adding benchmark and regression tests, adding confidences and provenances, and handling integrations for the Adept system, particularly in adding our output to the Adept KB.

3.6.4 Impact

Stanford software, principally CoreNLP, was an important part of the BBN Adept repository. It was used in the integrated Adept demonstration system and it was used by a number of other teams in their own software components. Stanford software was also used directly by the Research Innovation Group at DTRA (with NLP work done by MITRE) as part of DTRA J9CXQ – US SOCOM CWMD-T Support Program and by people in other parts of DoD. CoreNLP has moreover become a mainstay of basic linguistic processing that is also used by many researchers, startups and large companies.

4 RESULTS AND DISCUSSION

4.1 KBP SLOT FILLING AND COLD START KNOWLEDGE BASE POPULATION

During the grant, we took part each year in the annual Text Analysis Conference Knowledge Base Population (TAC KBP) evaluations. In early years, we did the Slot Filling (SF) or relation extraction task, just on English. Towards the end, the emphasis had shifted to Cold Start (CS) KBP and evaluation of the knowledge base, though SF results were also still calculated, and evaluation over all of English, Chinese, and Spanish. Once the task shifted to CS KBP, then a system also had to do entity linking (EL) to the knowledge base, and we used some of our own work in that area, but mainly depended on systems from colleagues.

4.1.1 Experience prior to this project

Stanford participated in the KBP Slotfilling task between 2009 and 2011. The submitted system placed 5th in 2009, below 5th in 2010, and 4th in 2011. We did not participate in 2012 but developed some strong new ideas for distantly supervised relation extraction published in Surdeanu et al. (2012).

4.1.2 TAC KBP 2013

Stanford's 2013 KBP entry achieved an F1 of 31.36 on the 2013 evaluation data, performing above the median entry (15.32 F1); see Angeli et al. (2013).

Stanford submitted 5 systems for the official evaluation. For all runs, a fixed confidence threshold of 0.5 was imposed based on the tuned threshold on the 2012 data. Slot fills under this threshold were discarded, with the exception of inferred slots which were always kept. In general, the first system S1 used everything we thought was best, and S2–5 either deleted components or added components that we were doubtful of the utility of. The five systems in Table 1 were:

- S1: The reference run, incorporating every component except inference using ReVerb relations.
- S2: S1, but with all the relation inference components disabled.
- S3: S1, but with experimental ReVerb OpenIE (Fader et al. 2011) inference paths enabled.
- S4: S1, but with inference, sentence-level competition, and additionally the hand-coded rules disabled. This run represents our system run with only MIML-RE and basic consistency.
- S5: S1, but using only the 2013 docs for searching for slot fills at test time. Thus, the component of our system which searches for provenance given a slot fill found in another corpus is not relevant.

The expected best system is S1; S2 removes inference; S3 adds ReVerb entailment rules; S4 removes all inference and rules, relying only on MIML-RE; S5 is identical to S1 but run only over the official 2013 source documents.

Table 1: Stanford's KBP SF submissions for 2013

2013 SYSTEMS	PRECISION	RECALL	F1
S1	35.8	27.9	31.4
S2	35.9	28.4	31.7
S3	35.1	26.7	30.3
S4	35.3	25.6	26.7
S5	38.2	26.7	31.5

Table 2: Stanford's 2013 KBP results as compared to other teams that year

2013 SYSTEMS	PRECISION	RECALL	F1
MEDIAN TEAM	15.0	15.7	15.3
STANFORD 2013	35.8	27.9	31.4
TOP TEAM	42.5	32.2	37.3

The results in Table 2 mean that we found only about one quarter of the relations that were found by someone (either another team or humans searching the text collection, and of the putative relations that we did find, only slightly more than one third of them were correct. This is still rather below the desirable accuracy of a textual relation extraction system! The conclusions we drew from these experiments and various side studies are as follows. The errors from incorrectly deduplicating entries would be helped by incorporating an entity linking system. The second class of errors – from not finding a sentence which adequately expresses the target relation – we thought we could address by improving our inference component to collect better weights for inferential paths, and to perform more holistic inference on the entity graph at test time with Markov Logic. The third class of errors – incorrect relation predictions – we thought we could mitigate by collecting crowd-sourced labels for the latent variables in MIML-RE. In part, this would provide valuable high-quality supervised training data, and in part it could make the model's objective more convex and manageable. We believed that the relatively small loss incurred from using our IR versus the Gold IR implies that our IR system performs well enough that it is not a bottleneck in improving performance on the task.

We achieved 6th place in the 2013 KBP slot filling evaluation. This was not only an enormous improvement over our last KBP Slot Filling evaluation performance (in 2011) but placed us among the top cluster of 6 systems with good (over 30% F1) scores, a group including only one other team from the U.S.A.

4.1.3 TAC KBP 2014

The second-year evaluation maintained use of our traditional system, using IR and MIML-RE, but we did some ensemble runs and joint development with DeepDive systems from Chris Ré's group. All the systems used NLP analyses produced by CoreNLP. The system is described in detail in Angeli et al. (2014). The results are shown in Table 3.

Table 3: Stanford's 2014 KBP results

2014 SYSTEMS	PRECISION	RECALL	F1
DEEPDIVE (HIGH RECALL)	54.4	27.8	36.8
DEEPDIVE (HIGH PRECISION)	54.8	24.9	34.3
MIML-RE	36.2	28.7	32.0
MIML-RE (– PATTERNS)	35.3	26.3	30.2
MIML-RE (– ACTIVE LEARNING)	29.2	26.2	27.6

The submitted systems were:

- DeepDive (high recall): A run using DeepDive, tuned along the calibration curve to tune for the best F1 and correspondingly relatively biased towards recall.
- DeepDive (high precision) A run using DeepDive, tuned along the calibration curve to favor higher precision (90% precision cutoff).
- MIML-RE The MIML-RE system, as in the 2013 submission, but with the addition of learned patterns and the new model from Angeli et al. (2014).
- MIML-RE (–patterns) The MIML-RE system, with all patterns removed. This is the expected performance of a system based entirely and only on MIML-RE and a coreference-based alternate names detector.
- MIML-RE (–active) The MIML-RE system, with the model from Stanford's 2013 submission

The precision of the DeepDive system is very impressive, but as the task is set up, TAC KBP remains largely a recall-bound task, and the high recall DeepDive system only modestly enhances recall. Our MIML-RE system achieves slightly better recall again, and so it remains not that far behind the DeepDive system in performance. Overall, the DeepDive high recall system was the strongest submitted system, and the MIML-RE system alone could have taken third place.

4.1.4 TAC KBP 2015

TAC KBP 2015 was the first outing of our third-generation relation database–backed KBP system. This system is described in detail in Angeli et al. (2015). In addition, it marked a period when we started to move away from distant supervision towards the use of crowd-source supervision and bootstrapping self-training methods. Our final system was an ensemble of a number of relation extractors, including our new work in neural and OpenIE extractors. The task also became more nuanced this year: systems were evaluated not only on slot fills (classic relation extraction) but on constructing a knowledge base, evaluated by looking at path length 1 (hop 0) and path length 2 (hop 1) queries on the knowledge base. In the results in this report, shown in Table 4, we show Hop All (averaged over hop 0 and hop 1) results computed directly from the knowledge base (which has to observe consistency constraints, unlike independent slot fills). This is the most rigorous evaluation condition and numbers go down considerably versus last year (for obvious reasons). Results on other measures more similar to previous years showed that we had considerably improved the precision of individual slot fills in our system, while largely maintaining recall. A skew towards precision seemed desirable, to maximize the chances of getting path length 2 queries right, but the recall of those became rather low.

The two systems submitted for KB evaluation were:

- Stanford1 A high precision system (patterns, openie, website, altnames) for both hop0 and hop1.
- Stanford2 A high recall system: the union of all models (adding supervised logistic regression and neural classifiers) for both hop0 and hop1.

Stanford's system basically matched the score of the other leading system (BBN), outperforming all other teams on Cold Start Hop 1 F_1 (i.e., results on answering questions like "where do people in Gaithersburg work?").

Table 4: KBP 2015 system results on Hop All KB evaluation

2015 SYSTEMS	PRECISION	RECALL	F1
STANFORD1 (HIGH PRECISION)	48.7	9.1	15.4
STANFORD2 (HIGH RECALL)	21.0	23.2	22.1

An interesting note about the 2015 evaluation was a comparison of machine performance to human performance on slot filling (as opposed to whole knowledge base construction). The evaluation included the performance of humans, who were given a limited time budget for searching for relevant information to extract to complete a slot fill. While overall human performance was better, due to their much higher precision, our high recall system achieved better fact recall than the humans, and because of that, was not too behind the humans in F1 score (the Stanford system got about 31% recall versus about 25% recall for the humans, still losing out at about 28% F1 versus 37% F1 for the humans).

4.1.5 TAC KBP 2016

This was the first year we attempted multilingual KBP, doing English and Chinese. Full details of our system this year can be found in Zhang et al. (2016). In Table 5, we show our official scores on the CS KBP track for the languages we took part in and the overall cross-lingual evaluation. Our English score showed no progress from the year before but remained the leading system. Our own testing showed that our system was considerably better than last year's system; it appeared that the evaluation set was just more difficult. Our first cut at a Chinese system was well below the performance of our English system, partly due to the difficulty of the language and partly due to us having fewer resources for Chinese and less development time. Our cross-lingual results were further dragged down by the fact that we did not have a Spanish system. Note also that because of the way the Hop All results were calculated by combining Hop-0 and Hop-1 scores, the F1 measure for Hop All need not fall between Precision and Recall; this phenomenon is observed for English.

Table 5: Macro-averaged LDC-MEAN KBP 2016 KB track Hop All scores

2016 SYSTEMS	PRECISION	RECALL	F1
STANFORD ENGLISH	22.1	25.9	22.0
STANFORD CHINESE	12.7	20.4	14.2
STANFORD CROSS-LINGUAL	10.9	13.9	11.2

4.1.6 TAC KBP 2017

For TAC KBP 2017 we submitted the only independent trilingual team doing Chinese, English, and Spanish relation extraction. We also participated in the Tinkerbell system with our colleagues. Our scores are as in Table 6, which were again the highest submitted scores. The English and Chinese scores show some nice progress, and our inaugural Spanish system is about as good as the Chinese system. We remain a little perplexed by the low cross-lingual score, thinking that something may have gone wrong in cross-lingual entity linking.

Table 6: Macro-averaged LDC-MEAN KBP 2017 KB track Hop All scores

2017 SYSTEMS	PRECISION	RECALL	F1
STANFORD ENGLISH	23.8	33.3	25.4
STANFORD CHINESE	19.6	18.1	18.0
STANFORD SPANISH	19.2	19.8	18.6
STANFORD CROSS-LINGUAL	12.9	13.3	11.7

4.2 OTHER RESULTS

Within the work of various particular research projects, we also did many other detailed evaluations of particular system components. These are not all reproduced in this report, and readers are referred to individual papers appearing in the references. However, in Table 7, we do show one cumulative table showing the progress of our scores for coreference resolution. The figures shown in the table are for the CoNLL 2012 coreference score, which is an average of three scores that had previously been used to score coreference systems: MUC, B^3, and CEAF-φ_4. There are differences between the scores, but, nevertheless, a score of 100 would mean getting all mention coreference decisions right. Because coreference resolution is essentially a mention clustering task, and the evaluation is of complete clusters, a few mistakes can greatly bring down the score. In some part coreference scores are low because it is a hard task, but in considerable measure the lower numbers are also due to this cluster-style evaluation, whereas several other NLP tasks (such as parsing and POS tagging) are conventionally evaluated at the level of individual decisions. In the table, bold picks out all of: the best CoNLL shared task scores; the best systems prior to the Stanford 2016 systems, and the best systems overall for each language.

Table 7: Coreference resolution CoNLL score on CoNLL 2012 test set

MODEL	ENGLISH	CHINESE
Stanford (Lee et al. 2011) [CoNLL 2011 winner]	57.80	—
Chen & Ng (2012) [CoNLL 2012 Chinese winner]	54.52	**57.63**
Fernandes (2012) [CoNLL 2012 English winner]	**60.65**	51.46
Björkelund & Kuhn. (2014) [Best previous Chinese system]	61.63	**60.06**
Stanford (Clark & Manning, ACL 2015)	63.02	—
Wiseman et al. (2016) [Best previous English system]	**64.21**	—
Stanford (Clark & Manning 2016a)	65.29	63.66
Stanford Deep RL Mention-Ranking (Clark & Manning 2016b)	**65.73**	**63.88**

5 CONCLUSIONS

Our major goal was to innovate on new methods for text understanding and knowledge extraction. The project did important work in developing the use of deep learning methods for natural language understanding and in developing new, improved algorithms for textual relation extraction and coreference resolution. While full text understanding is still far from a solved problem, the main goals of the project were achieved. Our group produced a variety of new and highly influential algorithms. During the early years of the project, our group produced much of the most cited work in using deep learning for natural language understanding, before use of these techniques disseminated more broadly. Our algorithms posted state-of-the-art results on a number of domains and tasks, and, partly through our making our algorithms broadly available in an integrated fashion through our CoreNLP software framework, they have had a considerable influence. The algorithms have seen considerable use, by many people in academia, government, the military, and industry. Our systems were adapted to handle both formally written sources like newspaper articles and from informal sources such as web forums. The system was extended to work in multiple languages, with our work covering English, Spanish, and Chinese. Overall, our work went some distance to showing that it was practical to automatically populate a knowledge base from a collection of raw text documents. Nevertheless, there remain many issues where further work is likely needed for robust deployment to be possible. These include all the well-known cases of NLP errors, ranging from mistakes in linking textual entity mentions to knowledge base entities, failures in entity mention recognition and correct parsing of text, and sins of omission and commission in asserting relations between entities. Nevertheless, our new generation of mainly neural network–based tools have brought NLP systems to a new level of performance.

6 REFERENCES

Gabor Angeli and Christopher Manning. 2013. Philosophers are Mortal: Inferring the Truth of Unseen Facts. In *CoNLL*.

Gabor Angeli, Arun Chaganty, Angel Chang, Kevin Reschke, Julie Tibshirani, Jean Y. Wu, Osbert Bastani, Keith Siilats, and Christopher D. Manning. 2013. Stanford's 2013 KBP System. In *TAC 2013: Text Analysis Conference Proceedings*.

Gabor Angeli and Chris D. Manning. 2014. NaturalLI: Natural Logic Inference for Common Sense Reasoning. In *Proceedings of EMNLP*.

Gabor Angeli, Sonal Gupta, Melvin Johnson Premkumar, Chris Manning, Chris Re, Julie Tibshirani, Jean Y. Wu, Sen Wu, and Ce Zhang. 2014. Stanford's Distantly Supervised Slot Filling Systems for KBP 2014. In *TAC 2014: Text Analysis Conference Proceedings*.

Gabor Angeli, Julie Tibshirani, Jean Y. Wu and Christopher D. Manning. 2014. Combining Distant and Partial Supervision for Relation Extraction. In *Proceedings of EMNLP*.

Gabor Angeli, Melvin Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging Linguistic Structure for Open Domain Information Extraction. In *Proceedings of the ACL*.

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Chaganty, Jason Bolton, Melvin Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Chris Manning. 2015. Bootstrapped Self Training for Knowledge Base Population. In *TAC 2015: Text Analysis Conference Proceedings*.

Gabor Angeli, Neha Nayak, and Christopher Manning. 2016. Combining Natural Logic and Shallow Reasoning for Question Answering. In *Proceedings of the ACL*.

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 & Interoperability with Discourse*, pages 178–186.

van Benthem, Johan. 2014. A Brief History of Natural Logic. In Chakraborti, M.K., Löwe, B., Mitra, M.N., Sarukkai, S. (eds.) Logic, Navya-Nyāya & Applications, Homage to Bimal Krishna Matilal. College Publications, London.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of EMNLP 2013*.

Jonathan Berant and Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *Proceedings of the ACL*.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In Association of Computational Linguistics (ACL), pages 47–57.

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Arun Tejasvi Chaganty, Ashwin Pradeep Paranjape, Percy Liang and Christopher D. Manning. 2017. Importance sampling for unbiased on-demand evaluation of knowledge base population. In *Proceedings of EMNLP*.

Angel X. Chang, Manolis Savva, and Christopher D Manning. 2014. Learning Spatial Knowledge for Text to 3D Scene Generation. In Empirical Methods in Natural Language Processing (EMNLP 2014).

Angel X. Chang, Will Monroe, Manolis Savva, Christopher Potts, and Christopher D. Manning. 2015. Text to 3D Scene Generation with Rich Lexical Grounding. In *Proceedings of the ACL*.

Chen Chen and Vincent Ng. 2012. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning: Shared Task*, pages 56–63.

Danqi Chen and Chris D. Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of EMNLP.*

Danqi Chen, Jason Bolton, Christopher D. Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Proceedings of the ACL.*

Kevin Clark and Christopher D. Manning. 2015. Entity-Centric Coreference Resolution with Model Stacking. In *Proceedings of the ACL*.

Kevin Clark and Christopher D. Manning. 2016a. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In *Proceedings of the ACL*.

Kevin Clark and Christopher D. Manning. 2016b. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In *Proceedings of EMNLP*.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task. In *Proceedings of CoNLL 2017.*

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP*.

Eraldo Rezende Fernandes, Cícero Nogueira Dos Santos, and Ruy Luiz Milidiu. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning: Shared Task*, pages 41–48.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. 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*, pages 28–34.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: Tools for querying and manipulating tree data structures. In *Proceedings of LREC.*

Minh-Thang Luong, Richard Socher, and Christopher Manning. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *Proceedings of CoNLL 2013.*

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL 2016.*

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Stephen J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL 2014 Demonstrations Session.*

Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *Proceedings of the ACL*.

Panupong Pasupat and Percy Liang. 2016. Inferring Logical Forms from Denotations. In Proceedings of the ACL.

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

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher D. Manning, and Daniel Jurafsky. 2014. Event Extraction Using Distant Supervision. In *Proceedings of LREC 2014*.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng and Christopher D. Manning. 2011. Parsing Natural Scenes and Natural Language With Recursive Neural Networks. In *International Conference on Machine Learning (ICML).*

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. 2013a. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of EMNLP 2013*.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013b. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In *Proceedings of NIPS*.

Richard Socher, John Bauer, Christopher Manning, and Andrew Ng. 2013c. Parsing with Compositional Vector Grammars. In *Proceedings of ACL 2013.*

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013d. Zero Shot Learning Through Cross-Modal Transfer. In *Proceedings of NIPS*.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded Compositional Semantics for Finding and Describing Images with Sentences. In *Transactions of the Association for Computational Linguistics (TACL), Vol. 2.*

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations: From Tree-Structured Long Short-Term Memory Networks.

Vivek Srikumar and Dan Roth. 2011. A joint model for extended semantic role labeling. In *Proceedings of EMNLP.*

Vivek Srikumar and Christopher D. Manning 2014. Learning Distributed Representations for Structured Output Prediction. In Proceedings of *NIPS 2014.*

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS 2014.*

Julie Tibshirani and Christopher D. Manning. 2014. Robust Logistic Regression using Shift Parameters. In *Proceedings of the Association for Computational Linguistics*.

Yushi Wang, Jonathan Berant and Percy Liang. 2015. Building a Semantic Parser Overnight. In *Proceedings of the ACL*.

Keenon Werling, Gabor Angeli, and Christopher D. Manning. 2015. Robust Subgraph Generation Improves Abstract Meaning Representation. In *Proceedings of the ACL*.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Human Language Technology and North American Association for Computational Linguistics (HLT-NAACL).*

Sen Wu, Ce Zhang, Christopher De Sa, Jaeho Shin, Feiran Wang, and C. Ré. 2015. Incremental Knowledge Base Construction Using DeepDive. In *VLDB 2015*.

Ce Zhang, Christopher Ré, Michael Cafarella, Christopher De Sa, Alex Ratner, Jaeho Shin, Feiran Wang, and Sen Wu. 2017. DeepDive: Declarative Knowledge Base Construction. *Communications of the ACM* 60(5): 93–102.

Yuhao Zhang, Arun Chaganty, Ashwin Paranjape, Danqi Chen, Jason Bolton, Peng Qi, and Christopher D. Manning. 2016. Stanford at TAC KBP 2016: Sealing Pipeline Leaks and Understanding Chinese. In *TAC 2016: Text Analysis Conference Proceedings*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of EMNLP*.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

CCG	Combinatory Categorial Grammar
CKY	Cocke-Kasami-Younger. The basic O(n^3) context-free grammar parsing technique
CoNLL	The Conference on Natural Language Learning (and its shared tasks)
CS	Cold Start
CVG	Compositional Vector Grammar (a grammar over real-valued representations)
DEFT	Deep Exploration and Filtering of Text (the DARPA program behind this work)
EL	Entity Linking
F1	Equally weighted harmonic mean of precision and recall
GPE	Geopolitical entity (a named entity recognition class)
IR	Information retrieval
JVM	Java Virtual Machine
KB	Knowledge Base
KBC	Knowledge Base Completion
KBP	Knowledge Base Population
LSTM	Long short-term memory (an effective sort of neural sequence model cell)
MIML	Multiple-instance, multiple-label
MIML-RE	Multiple-instance, multiple-label relation extraction
NaturalLI	Natural Logic Inference (a system using natural logic for KBC)
NER	Named Entity Recognition
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NN	Neural Net
NTN	Neural Tensor Network
OpenIE	Open Information Extraction
PCFG	Probabilistic Context-Free Grammar
POS	Part-of-Speech (tagging)
RNTN	Recursive Neural Tensor Network
SF	Slot Filling
TAC	Text Analysis Conference
TACRED	TAC Relation Extraction Dataset
TreeRNN	Tree Recursive Neural Network
UD	Universal Dependencies (a common annotation framework for languages)