UNDERSTANDING STORIES VIA EVENT SEQUENCE MODELING

BY

HAORUO PENG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Dan Roth, Chair
Professor Julia Hockenmaier
Professor Jian Peng
Professor Kevin Gimpel, Toyota Technological Institute at Chicago

Abstract

Understanding stories, i.e. sequences of events, is a crucial yet challenging natural language understanding (NLU) problem, which requires dealing with multiple aspects of semantics, including actions, entities and emotions, as well as background knowledge. In this thesis, towards the goal of building a NLU system that can model what has happened in stories and predict what would happen in the future, we contribute on three fronts: First, we investigate the optimal way to model events in text; Second, we study how we can model a sequence of events with the balance of generality and specificity; Third, we improve event sequence modeling by joint modeling of semantic information and incorporating background knowledge.

Each of the above three research problems poses both conceptual and computational challenges. For event extraction, we find that Semantic Role Labeling (SRL) signals can be served as good intermediate representations for events, thus giving us the ability to reliably identify events with minimal supervision. In addition, since it is important to resolve co-referred entities for extracted events, we make improvements to an existing co-reference resolution system. To model event sequences, we start from studying within document event co-reference (the simplest flow of events); and then extend to model two other more natural event sequences along with discourse phenomena while abstracting over the specific mentions of predicates and entities. We further identify problems for the basic event sequence models, where we fail to capture multiple semantic aspects and background knowledge. We then improve our system by jointly modeling frames, entities and sentiments, yielding joint representations of all these semantic aspects; while at the same time incorporate explicit background knowledge acquired from other corpus as well as human experience. For all tasks, we evaluate the developed algorithms and models on benchmark datasets and achieve better performance compared to other highly competitive methods.

Publication Notes

Parts of the work in this thesis have appeared in the following publications:

- Haoruo Peng, Snigdha Chaturvedi, and Dan Roth. A joint model for semantic sequences: Frames, entities, sentiments. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, 2017.
- Haoruo Peng, Yangqiu Song, and Dan Roth. Event detection and co-reference with minimal supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- Haoruo Peng and Dan Roth. Two discourse driven language models for semantics. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.
- Haoruo Peng, Kai-Wei Chang, and Dan Roth. A joint framework for coreference resolution and mention head detection. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, 2015.
- Haoruo Peng, Daniel Khashabi, and Dan Roth. Solving hard coreference problems. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.
- Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. Improving temporal relation extraction with a globally acquired statistical resource. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.
- Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. Story comprehension for predicting what happens next. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.
- Chase Duncan, Liang-Wei Chan, Haoruo Peng, Hao Wu, Shyam Upadhyay, Nitish Gupta, Chen-Tse Tsai, Mark Sammons, and Dan Roth. Illinois CCG TAC 2017 entity discovery and linking, and event

nugget detection and co-reference systems. In *Proceedings of the Text Analysis Conference (TAC)*, 2017.

- Chen-Tse Tsai, Stephen Mayhew, Haoruo Peng, Mark Sammons, Bhargav Mangipundi, Pavankumar Reddy, and Dan Roth. Illinois CCG TAC 2016 entity discovery and linking, event nugget detection and co-reference, and slot filler validation systems. In *Proceedings of the Text Analysis Conference (TAC)*, 2016.

- Mark Sammons, Haoruo Peng, Yangqiu Song, Shyam Upadhyay, Chen-Tse Tsai, Pavankumar Reddy, Subhro Roy, and Dan Roth. Illinois CCG TAC 2015 event nugget, entity discovery and linking, and slot filler validation systems. In *Proceedings of the Text Analysis Conference (TAC)*, 2015.

- Yangqiu Song, Haoruo Peng, Parisa Kordjamshidi, Mark Sammons, and Dan Roth. Improving a pipeline architecture for shallow discourse parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL) - Shared task*, 2015.

To Father and Mother.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Prof. Dan Roth, for all the supports in the last five years. I really enjoy solving challenging problems with him and I always admire his insightful ideas. He selects good research directions, provides me with adequate resources, removes any obstacle in the way, and more importantly, he gives me enough freedom to explore problems according to our own interests. Besides critical thinking, I learned how to discuss, listen, debate, and present from him. He demonstrates what is a remarkable scholar and a passionate educator.

I would like to thank my committee members: Prof. Julia Hockenmaier, Prof. Jian Peng and Prof. Kevin Gimpel, from whom I got valuable feedback on my research. I would also like to thank all my internship mentors and collaborators, namely Samer Hassan, Ming-Wei Chang, Scott Yih, Zhongyuan Wang, Haixun Wang; I have learned a lot working with them.

I would like to thank everyone I encountered in the last five years. They shaped me and made my life in Champaign-Urbana more colorful. I especially acknowledge the members in the CogComp group. In my first two years, Kai-Wei Chang and Yangqiu Song helped me fitting into the group and understanding the research area more quickly. Mark Sammons and Eric Horn assisted me in many aspects such as software and traveling issues. Moreover, many thanks to other colleagues and collaborators, Daniel Khashabi, Snigdha Chaturvedi, Shyam Upadhyay, Qiang Ning, Chao Zhang, Subhro Roy, Nitish Gupta, Shashank Gupta, Stephen Mayhew, Chase Duncan, Christos Christodoupoulos, Hao Wu, Pavan Muddireddy, John Wieting. We had great discussions and I definitely have learned a lot from you.

Finally, I would like to thank my parents, for their unconditional sacrifice and support. I will not be able to go this far without you.

Table of Contents

List of Tables . . . . . **x**

List of Figures . . . . . **xiv**

Chapter 1 Introduction **1**
1.1 Motivation 1
1.2 Challenges 3
1.3 Thesis Contributions 5
1.4 Thesis Overview 5

Chapter 2 Background **8**
2.1 Related Tasks 8
2.1.1 Event Extraction and Co-reference 9
2.1.2 Entity Co-reference Resolution 10
2.1.3 Shallow Discourse Parsing 11
2.1.4 Script Learning 12
2.2 Related Techniques 13
2.2.1 Dataless Classification 13
2.2.2 Integer Linear Program 14
2.2.3 Neural Language Models 14

Chapter 3 Event Extraction **16**
3.1 Overview of Minimally Supervised Event Pipeline 16
3.2 Structured Vector Representation 18
3.3 Text-Vector Conversion 20
3.4 Event Mention Detection 21
3.5 Experiments 22

Chapter 4 Entity Coreference Resolution **25**
4.1 Joint Framework for Mention Head Detection and Co-reference 25
4.1.1 Motivation 25
4.1.2 System Design 25
4.1.3 Mention Head Candidate Generation 29
4.1.4 Mention Head Detection 30
4.1.5 ILP-based Mention-Pair Coreference 30
4.1.6 Joint Inference Framework 31
4.1.7 Joint Learning Framework 31
4.1.8 Stochastic Subgradient Descent for Joint Learning 32
4.1.9 Experiments 33
4.2 Solving Hard Co-reference Problems 37
4.2.1 Motivation 37
4.2.2 System Design 37

4.2.3 Predicate Schema . . . 39
4.2.4 Constrained ILP Inference . . . 41
4.2.5 Knowledge Acquisition . . . 43
4.2.6 Experiments . . . 45

Chapter 5 Event Coreference Resolution . . . 49
5.1 System Design . . . 49
5.2 Experiments . . . 50
5.2.1 Experimental Setup . . . 50
5.2.2 Empirical Results . . . 50

Chapter 6 Semantic Language Models . . . 53
6.1 Motivation . . . 53
6.2 Two Basic Semantic Language Models . . . 55
6.2.1 Semantic Frames and Discourse Markers . . . 55
6.2.2 Frame-Chain SemLM . . . 56
6.2.3 Entity-Centered SemLM . . . 56
6.3 Implementation of SemLMs . . . 57
6.3.1 N-gram Model . . . 57
6.3.2 Skip-Gram Model . . . 57
6.3.3 Continuous Bag-of-Words Model . . . 58
6.3.4 Log-bilinear Model . . . 58
6.4 Build SemLMs from Scratch . . . 58
6.4.1 Dataset and Preprocessing . . . 59
6.4.2 Semantic Unit Generation . . . 59
6.4.3 Language Model Training . . . 60
6.5 Evaluations . . . 61
6.5.1 Quality Evaluation of SemLMs . . . 61
6.5.2 Evaluation of SemLM Applications . . . 65

Chapter 7 SemLM with Multiple Semantic Aspects . . . 68
7.1 Motivation . . . 68
7.2 Semantic Aspect Modeling . . . 70
7.2.1 Semantic Frames . . . 71
7.2.2 Entities . . . 71
7.2.3 Sentiments . . . 72
7.3 FES-LM - Joint Modeling . . . 72
7.3.1 FES Representation . . . 73
7.3.2 Neural Language Model . . . 73
7.4 Building FES-LM . . . 74
7.4.1 Dataset and Preprocessing . . . 74
7.4.2 FES Representation Generation . . . 75
7.4.3 Neural Language Model Training . . . 76
7.5 Evaluation . . . 77
7.5.1 Quality of FES-LM . . . 78
7.5.2 Application on News . . . 78
7.5.3 Application on Stories . . . 79
7.5.4 Qualitative Analysis . . . 81

Chapter 8 SemLM with Knowledge **83**
8.1 Motivation 83
8.2 Event and Knowledge Modeling 86
8.2.1 Event Representation 86
8.2.2 Knowledge: Causality between Events 87
8.3 Knowledge Infused SemLM 89
8.3.1 FES-RNNLM 89
8.3.2 `KnowSemLM` 90
8.4 Building `KnowSemLM` 91
8.4.1 Dataset and Preprocessing 92
8.4.2 Event Abstractions 92
8.4.3 Knowledge Mining 93
8.4.4 Model Training 94
8.5 Experiments 94
8.5.1 Application for Story Prediction 94
8.5.2 Application for Referent Prediction 96
8.5.3 Analysis of `KnowSemLM` 97

Chapter 9 Conclusion **99**
9.1 Summary 99
9.2 Future Directions 100

References **101**

Appendix
Raw Text of Event Seeds **112**

List of Tables

3.1 Comparing requirements of MSEP and other methods. Supervised methods need all three resources while MSEP only needs an annotation guideline (as event examples). 17
3.2 Semantic role labeling coverage. We evaluate both "Predicates over Triggers" and "SRL Arguments over Event Arguments". "All" stands for the combination of Verb-SRL and Nom-SRL. The evaluation is done on all data. 19
3.3 Statistics for the ACE and TAC-KBP corpora. #Sent. is the number of sentences, #Men. is the number of event mentions, and #Cluster is the number of event clusters (including singletons). Note that the proposed MSEP does not need any training data. ACE(Test) is only used to evaluate event detection while we do cross-validation for ACE event co-reference. TAC-KBP(Test) is used for both event detection and co-reference evaluations. 23
3.4 Event Extraction (trigger identification) results. 23
3.5 Domain Transfer Results. We conduct the evaluation on TAC-KBP corpus with the split of newswire (NW) and discussion form (DF) documents. Here, we choose MSEP-EMD for event detection. We use SSED as the supervised module for comparison. We compare F1 scores of span plus type match. 24

4.1 Performance gaps between using gold mentions and predicted mentions. Performance gaps are always larger than 10%. Illinois's system (Chang et al., 2013) is evaluated on CoNLL (2012, 2011) Shared Task and ACE-2004 datasets. It reports an average F1 score of MUC, B^3 and $CEAF_e$ metrics using CoNLL v7.0 scorer. Berkeley's system (Durrett and Klein, 2013) reports the same average score on the CoNLL-2011 Shared Task dataset. Results of Stanford's system (Lee et al., 2011) are for B^3 metric on ACE-2004 dataset. 26
4.2 Performance of coreference resolution on the ACE-2004 and CoNLL-2012 dataset. Subscripts ($_M$, $_H$) indicate evaluations on (mentions, mention heads) respectively. For gold mentions and mention heads, they yield the same performance for coreference. Our proposed *H-Joint-M* system achieves the highest performance as in June 2015. Parameters of our proposed system are tuned as $\alpha = 0.9$, $\beta = 0.9$, $\lambda_1 = 0.25$ and $\lambda_2 = 0.2$. We also include more recent results where the current state-of-the-art performance is achieved by Peters et al. (2018) using ELMo embeddings. 34
4.3 Performance of mention detection on the ACE-2004 and CoNLL-2012 datasets. Subscripts ($_M$, $_H$) indicate evaluations on (mentions, mention heads) respectively. 36
4.4 Analysis of performance improvement in terms of *Mention Detection Error Reduction* (MDER) and *Performance Gap Reduction* (PGR) for coreference resolution on the ACE-2004 and CoNLL-2012 datasets. "Head" represents the mention head candidate generation module, "Joint" represents the joint learning and inference framework, and "H-Joint-M" indicates the end-to-end system. 36
4.5 Example sentences for each schema category. The annotated entities and pronouns are hard coreference problems. 39
4.6 Predicate Schemas and examples of the logic behind the schema design. Here $*$ indicates that the argument is dropped, and $\mathcal{S}(.)$ denotes the scoring function defined in the text. 39
4.7 Possible variations for scoring function statistics. Here $*$ indicates that the argument is dropped. 40

4.8 Extracting the polarity score given polarity information of a mention-pair (u, v). To be brief, we use the shorthand notation $p_v \triangleq pred_v$ and $p_u \triangleq pred_u$. $\mathbf{1}\{\cdot\}$ is an indicator function. $s_{pol}(u, v)$ is a binary vector of size three. 43
4.9 Statistics of *Winograd*, *WinoCoref*, *ACE* and *OntoNotes* datasets. We give the total number of mentions and pronouns, while the number of predictions for pronoun is specific for the test data. We added 746 mentions (709 among them are pronouns) to *WinoCoref* compared to *Winograd*. 45
4.10 Summary of learning and inference methods for all systems. SF stands for schema features while SC represents constraints from schema knowledge. 47
4.11 Performance results on *Winograd* and *WinoCoref* datasets. All our three systems are trained on *WinoCoref*, and we evaluate the predictions on both datasets. Our systems improve over the baselines by over than 20% on *Winograd* and over 15% on *WinoCoref*. 47
4.12 Performance results on *ACE* and *OntoNotes* datasets. Our system gets the same level of performance compared to a state-of-art general coreference system. 47
4.13 Distribution of instances in *Winograd* dataset of each category. Cat1/Cat2 is the subset of instances that require Type 1/Type 2 schema knowledge, respectively. All other instances are put into Cat3. Cat1 and Cat2 instances can be covered by our proposed Predicate Schemas. . 48
4.14 Ablation Study of Knowledge Schemas on *WinoCoref*. The first line specifies the performance for *KnowComb* with only Type 1 schema knowledge tested on all data while the third line specifies the performance using the same model but tested on Cat1 data. The second line specifies the performance results for *KnowComb* system with only Type 2 schema knowledge on all data while the fourth line specifies the performance using the same model but tested on Cat2 data. 48

5.1 **Event Co-reference Results on Gold Event Triggers.** "MSEP-Coref$_{\text{ESA,BC,W2V,DEP}}$" are variations of the proposed MSEP event co-reference system using ESA, Brown Cluster, Word2Vec and Dependency Embedding representations respectively. "MSEP-Coref$_{\text{ESA+AUG}}$" uses augmented ESA event vector representation and "MSEP-Coref$_{\text{ESA+AUG+KNOW}}$" applies knowledge to detect conflicting events. (GA) means that we use gold event arguments instead of approximated ones from SRL. 51

6.1 Comparison of vocabularies between frame-chain (FC) and entity-centered (EC) SemLMs. "F-Sen" stands for frames with predicate sense information while "F-Arg" stands for frames with argument role label information; "Conn" means discourse marker and "Per" means period. "Seq/Doc" represents the number of sequence per document. 55
6.2 **Statistics on SemLM vocabularies and sequences.** "F-s" stands for single frame while "F-c" stands for compound frame; "Conn" means discourse marker. "#seq" is the number of sequences, and "#token" is the total number of tokens (semantic units). We also compute the average token in a sequence i.e. "#t/s". We compare frame-chain (FC) and entity-centered (EC) SemLMs to the usual syntactic language model setting i.e. "LM". 60
6.3 Perplexities for SemLMs. UNI, BG, TRI, CBOW, SG, LB are different language model implementations while "FC" and "EC" stand for the two SemLM models studied, respectively. "FC-FM" and "EC-FM" indicate that we removed the "FrameNet Mapping" step. Note that for CBOW, SG and LB models, the perplexity numbers are not directly comparable with the N-gram model. 62
6.4 Narrative cloze test results for SemLMs. UNI, BG, TRI, CBOW, SG, LB are different language model implementations while "FC" and "EC" stand for our two SemLM models, respectively. "FC-FM" and "EC-FM" mean that we remove the FrameNet mappings. "w/o DIS" indicates the removal of discourse makers in SemLMs. "Rel-Impr" indicates the relative improvement of the best performing SemLM over the strongest baseline. We evaluate on two metrics: mean reciprocal rank (MRR)/recall at 30 (Recall@30). LB outperforms other methods for both frame-chain and entity-centered SemLMs. 63

6.5 Co-reference resolution results with entity-centered SemLM features. "EC" stands for the entity-centered SemLM. "TRI" is the tri-gram model while "LB" is the log-bilinear model. "p_c" means conditional probability features and "em" represents frame embedding features. "w/o DIS" indicates the ablation study by removing all discourse makers for SemLMs. We conduct the experiments by adding SemLM features into the base system. We outperform the state-of-art system (Wiseman et al., 2015) (as in 2015). The improvement achieved by "EC_LB ($p_c + em$)" over the base system is statistically significant. We also include more recent results where the current state-of-the-art performance is achieved by Peters et al. (2018) using ELMo embeddings. 64
6.6 Shallow discourse parsing results with frame-chain SemLM features. "FC" stands for the frame-chain SemLM. "TRI" is the tri-gram model while "LB" is the log-bilinear model. "p_c", "em" are conditional probability and frame embedding features, resp. "w/o DIS" indicates the case where we remove all discourse makers for SemLMs. We do the experiments by adding SemLM features to the base system. The improvement achieved by "FC-LB ($p_c + em$)" over the baseline is statistically significant. 65

7.1 Comparison of generative ability for different models. For each model, we provide Ex.1 as context and compare the generated ending. 4-gram and RNNLM models are trained on NYT news data while Seq2Seq model is trained on the story data (details see Sec. 7.5). These are models operated on the word level. We compare them with FC-SemLM (Peng and Roth, 2016), which works on frame abstractions, i.e. "predicate.sense". For the proposed FES-LM, we further assign the arguments (subject and object) of a predicate with NER types ("PER, LOC, ORG, MISC") or "ARG" if otherwise. Each argument is also associated with a "[new/old]" label indicating if it is first mentioned in the sequence (decided by entity co-reference). Additionally, the sentiment of a frame is represented as positive (POS), neural (NEU) or negative (NEG). FES-LM can generate better endings in terms of soundness and specificity. The FES-LM ending can be understood as "[Something] convict a person, who has been mentioned before (with an overall negative sentiment)", which can be instantiated as "Steven Avery was convicted." given current context. 68
7.2 Statistics on FES-LM vocabularies and sequences. We compare FES-LM trained on NYT vs. ROCStories; "FES" stands for unique FES representations while "F" for frame embeddings, "E" for entity representations, and "S" for sentiment representations. "#seq" is the number of sequences, and "#token" is the total number of tokens (FES representations) used for training. 76
7.3 Quality comparison of neural language models. We report results for narrative cloze test. The evaluation is done on the gold PropBank data (annotated with gold frames). LBL outperforms CBOW and SG. We carry out ablation studies for FES-LM without entity and sentiment aspects respectively. 77
7.4 Shallow discourse parsing results. With added FES-LM features, we get significant improvement (based on McNemar's Test) over the base system(*) and outperform SemLM, which only models frame information. We also rival the top system (Mihaylov and Frank, 2016) in the CoNLL16 Shared Task (connective sense classification subtask). 77
7.5 Accuracy results for story cloze text in the unsupervised setting. "S." represents the inference method with the single most informative feature while "M.V." means majority voting. FES-LM outperforms the strongest baseline (Seq2Seq with attention) by 3 points. The difference is statistically significant based on McNemar's Test. Additional ablation studies show that each semantic aspect contributes to the joint model. 79

8.1 Accuracy results for story cloze test in the unsupervised setting. "S." represents the inference method with the single most informative feature while "M.V." means majority voting. `KnowSemLM` outperforms both the baselines and the base model without the use of knowledge. 95

8.2 Accuracy results for the referent prediction task on InScript Corpus. We re-implemented the base model (Modi et al., 2017) as "Re-base", and apply `KnowSemLM` to add additional features. "Re-Base w/ FES-RNNLM" is the ablation study where no event causality knowledge is used. Even though "Re-base" model performs not as good as the original base model, we achieve the best performance with added `KnowSemLM` features. 96
8.3 Results for Perplexity and Narrative Cloze Test. Both studies are conducted on the NYT hold-out data. "FES-RNNLM" represents as the semantic language model without the use of knowledge. The numbers shows that `KnowSemLM` has lower perplexity and higher recall on narrative cloze test; which demonstrates the contribution of the infused event causality knowledge. 97
8.4 Statistics for the use of event causality knowledge in `KnowSemLM`. We gather the statistics for both NYT and InScript Corpus. "Match/Event" represents average number of times a casual event match is found in the event causality knowledge base per event; while "Activation/Event" stands for the average number of times we actually generate event predictions from the outcome events of the knowledge base. In addition, we believe the ratio of "Activation/Event" over "Match/Event" co-relates with the scaling parameter λ. 98

List of Figures

3.1 An overview of the end-to-end MSEP system. "Event Examples" are the only supervision. No training is needed for MSEP. Event co-reference will be discussed in Chapter 5. 18
3.2 Basic event vector representation. Event vector is the concatenation of vectors corresponding to action, agent$_{sub}$, agent$_{obj}$, location, time and sentence/clause. 21
3.3 Augmented event vector representation. Event vector is the concatenation of vectors corresponding to basic event vector representation, agent$_{sub}$ + action, agent$_{obj}$ + action, location + action and time + action. Here, "+" means that we first put text fragments together and then convert the combined text fragment into an ESA vector. 21

4.1 Comparison between a traditional pipelined system and our proposed system. We split up mention detection into two steps: mention head candidate generation and (an optional) mention boundary detection. We feed mention heads rather than complete mentions into the coreference model. During the joint head-coreference process, we reject some mention head candidates and then recover complete mention boundaries after coreference decisions are made. 27

7.1 Examples of short stories requiring different aspects of semantic knowledge. For all stories, Opt.1 is the correct follow-up, while Opt.2 is the contrastive wrong follow-up demonstrating the importance of each aspect. Alter. showcases an alternative correct follow-up, which requires considering different aspects of semantics jointly. 69
7.2 Examples of the need for different levels of entity abstraction. For each sentence, one wants to understand what the pronoun "she" refers to, which requires different abstractions for two underlined entity choices depending on context. 71
7.3 An overview of the FES representation in a semantic sequence. Semantic frames are represented by vector r_f. The entity representation r_e is the concatenation of r_{sub} and r_{obj}, both consist of two parts: an one-hot vector for entity type plus an additional dimension to indicate whether or not it is a *new entity*. The sentiment representation r_s is also one-hot. 73
7.4 Examples of stories where FES-LM makes correct predictions for the ending. We use data from ROCStories dataset and predictions come from FES-LM with the inference method of single most informative feature. 80
7.5 Examples of stories where FES-LM fails to make correct predictions for the ending. We use data from ROCStories dataset and predictions come from FES-LM with the inference method of single most informative feature. 81

8.1 Local and Global Context Information when Modeling Event Sequences. Blue dots denote events already described in text. The blue circle indicates local context, i.e. events learned from a large corpus via language models; while the red circle represents global context, i.e. events learned from human experience via knowledge of event causality (which may have overlaps with local context). For event representations, we abstract over the surface forms of semantic frames and entities. The proposed `KnowSemLM` leverages both information to better predict future events. 84

8.2 Overview of the Computational Workflow for the proposed `KnowSemLM`. There are two key components: 1) a knowledge selection model, which activates the use of knowledge based on matching casual event and produce a distribution over outcome events via attention; 2) a sequence generation model, which takes input from both the knowledge selection model and the base semantic language model (FES-RNNLM) to generate future events via a copying mechanism. Note that the single dots indicate explicit event representations while three consecutive dots stand for event vectors. 90

Chapter 1

Introduction

It has long been a goal in Artificial Intelligence (AI) that we can build a system to *understand stories*. The success of such a system will potentially lead to a major breakthrough in AI, as well as many practical applications, such as reading comprehension (Hirschman et al., 1999; Richardson et al., 2013; Rajpurkar et al., 2016), conversational bots (Ritter et al., 2011), and machine translation (Brown et al., 1990). In this chapter, we specify the scope of *stories* that we study and also what constitutes as *understanding*.

1.1 Motivation

We put story understanding in the context of natural language understanding. In this dissertation, we deal with text input that is used to tell stories, e.g. news stories (relatively long) as well as scripts and narratives (relatively short). For example, the following paragraph describes a news story of a diplomatic *visit*.

> *Ambassador visits French researcher in Tehran prison*
> *PARIS, Aug 14, 2009 (AFP)*
> *France's ambassador to Iran on Friday visited a young French academic in the Tehran prison where she is being held on spying charges, the foreign ministry said here. "He explained to her that the French authorities are doing all they can to obtain her release as soon as possible," a spokesman said. The visit was ambassador Bernard Poletti's second trip to Evin prison to see Clotide Reiss, who was among at least 110 defendants tried last week on charges related to post-election protests.*

To fully understand the story, we need to correctly extract each of the events mentioned in this piece of text, i.e. "a French ambassador visited a French researcher", "the French ambassador talked to the prisoner", "the French academic was tried among others". In addition, we know that the name of the French ambassador is *Bernard Poletti* while the imprisoned French academic is called *Clotide Reiss*; the location of the visit is *an Tehran prison, namely Evin prison*; the time of the visit is *Friday* while the time of the trial is *last week* (relative to the time of the report, *Aug 14, 2009*). This shows that an AI system needs to accurately extract events from text, where each event contains the essential information on its actions, agents, location and time. In this process, it is also necessary to co-relate entities that refer to the same

thing (e.g. *French ambassador - He - Bernard Poletti*, *French academic - she - her - Clotide Reiss*). Thus, the AI system would need to have the ability to resolve entity co-reference.

Apart from understanding what has happened in the story, humans are able to predict what is likely to happen in the future based on both the context provided in text and his/her common sense knowledge gained from human experience. This constitutes another perspective of story understanding for AI systems: the ability to predict how likely a future event would happen based on given a given story (i.e. a sequence of events). For example, the following narrative story describes an incident happened to *Addie*.

> *Addie was <u>working</u> at the mall at Hollister when a strange man <u>came in</u>. Before she knew it, Addie <u>looked behind</u> her and <u>saw</u> stolen clothes. Addie <u>got scared</u> and <u>tried to chase</u> the man out. Luckily guards <u>came</u> and <u>arrested</u> him.*
> *Ending opt.1 - Addie was relieved and took deep breaths to calm herself.*
> *Ending opt.2 - Addie was put in jail for her crime.*

Given such context, it is obvious for humans to understand that "Addie was relieved and took deep breaths to calm herself" is a much more likely event, compared to "Addie was put in jail for her crime." In order to make such a prediction, an AI system should model the sequence of events happened in text (each underlined action indicates an event). Thus, a computational model can be built on top of such event sequences and we shall be able to not only assign probabilities of a future event, but also generate future event sequences. Before we can achieve this goal, we also study the simplest form of event sequence, i.e. event co-reference, which is a chain of events that describes the same thing. Moreover, for sequence modeling, it is also crucial to capture discourse information (frequently expressed by explicit discourse markers).

> *Sent.A1 - Kevin was robbed by Robert, <u>and</u> he was arrested by the police.*
> *Sent.A2 - Kevin was robbed by Robert, <u>but</u> the police mistakenly arrested him.*
>
> *Sent.B1 - Mark began the investigation, <u>before</u> the evidence became public knowledge.*
> *Sent.B2 - Mark began the investigation, <u>after</u> the evidence became public knowledge.*

In the above examples (A1 v.s. A2), the semantic meaning of the text after the discourse markers (and/but) is totally changed. In Sent.A1, "he" is referred to Robert since the *robber* is the one to be arrested by police after the robbery in a natural way. However, in Sent.A2, the discourse marker "but" indicates a disruption of such a natural order, where an unexpected event happened, i.e. the police arrested Kevin. Even more evidently, consider the semantic meanings of Sent.B1 and Sent.B2, they have different (judicial) implications, which may lead to different subsequent events and results of the "investigation". Both cases demonstrate the importance of discourse information when modeling event sequences.

In this thesis, we investigate the problems of extracting events from text, and predicting future events based on event sequences in context; which enables us to make progress for story understanding in NLP.

1.2 Challenges

We address several key challenges in this thesis. First, it is non-trivial to accurately extract events from text. There is no universally agreed event definition. In addition, the training data for event extraction is always limited across different settings. Second, modeling event sequences is not straightforward. We need to abstract over semantic frames and entities to achieve a balance of generality and specificity. Moreover, for cases where external knowledge is required, we study how we can incorporate it into event sequence modeling. The challenges we tackle in this dissertation is summarized below:

- No unified definition for events

  There is no commonly accepted definition of what an event is or which information is connected to an event. The Oxford dictionary (Stevenson, 2010) gives a rather broad definition for the word *event*: *A thing that happens or takes place, especially one of importance.* This is one definition for events, however, the definition of events is ambiguous and changes from area to area. Further, there is no agreed standard of which types of events exist. In the field of NLP, Jurafsky and Martin (2007) describe that understanding an event means to be able to answer the question: *who did what to whom and perhaps also when and where.* The answer to this question can be scattered across a sentence or document, and the same real-world event can be described in various ways. Further, according to Hovy et al. (2013), it is not clear whether our current set of events' definitions is adequate. Thus, given the complexity and fundamental difficulties, the current evaluation methodology in area of event extraction focuses on a limited domain of events, e.g. 33 types in ACE 2005 (NIST, 2005) and 38 types in TAC KBP (Mitamura et al., 2015).

  Since there is a parallel between event structures and sentence structures, we use outputs of Semantic Role Labeling (SRL) to approximate the each event component, e.g. event actions mostly correspond to predicates, event arguments can be roughly mapped to SRL arguments. In this dissertation, we show that SRL can be employed as a good intermediate representation for events.

- Limited amount of event training data

  There only exists a limited amount of training data for event extraction and event co-reference. Consequently, this allows researchers to train supervised systems that are tailored to these sets of events and

that overfit the small domain covered in the annotated data, rather than address the realistic problem of understanding events in text.

In the thesis, we pursue an approach that only requires minimal event supervision, which is in the form of a few event examples. We formulate event detection and co-reference as semantic relatedness problems, which we can scale to deal with a lot more types and also generalize across domains. Thus, the key challenge to solve becomes how to represent events, and how to model event similarity.

- Appropriate level of abstraction over entities and frames

  In order to model event sequences most efficiently, we need to balance between generality and specificity for events. Instead of modeling surface forms directly, we choose to abstract over entities and frames to achieve a higher level of generality. For example, we map both predicates "leave.01" and "depart.01"[1] to "Departing" based on FrameNet (Baker et al., 1998); we also represent entities like "John" and "Jack" as "Person". However, sometimes it is tricky to determine the correct level of abstraction for the purpose of preserving the complete semantic meaning of an event. Consider the following example:

  > *Case.A - <u>The doctor</u> told <u>Susan</u> that <u>she</u> had been busy.*
  > *Case.B - <u>The doctor</u> told <u>Susan</u> that <u>she</u> had cancer.*

  Here, in Case.A, it is enough to abstract both "The doctor" and "Susan" as "Person" since we can change both entities into other person names, and still have the semantic meaning of the sentence (one way to determine is that "she" would still refer to the subject of "tell"). However, in Case.B, it is important to know that the sentence describes an interaction between "doctor" and "patient" instead of just two people. Otherwise, we lose the necessary semantic information to correctly resolve "she" to "Susan" (because normally a "patient" gets the disease instead of a "doctor"). We address this problem by jointly modeling multiple aspects of the semantic information and discourse phenomena contained in events.

- Incorporation of common sense knowledge

  Humans' understanding of whether a specific event will occur or not depends not only on what has happened earlier in the sequence of events, but also on some "known" facts gained through human experience. For example,

  > *John first <u>checked in</u> at the counter, and then <u>went through</u> security checks.*

[1] "01" indicates the disambiguated predicate sense

We know from past experiences that travelers need to "check in" before "going through security checks" when taking a flight. When model such event sequences, current statistical learning models (e.g. language models) have significant limitation in the ability to encode and decode such factual knowledge. This is mainly because they cannot acquire such knowledge from statistical co-occurrences in the given text. Thus, we attempt to build a knowledge infused semantic language model which combines knowledge from external sources with the basic semantic language model trained from the given text corpus.

1.3 Thesis Contributions

In this thesis, we investigate each of the challenges mentioned above for story understanding. In particular, we make the following claim:

Story understanding can be better achieved via event sequence modeling, on an abstraction level of frames and entities instead of tokens.

The primary contributions of this dissertation are summarized below:

1. Improvement on Event Representations: We develop an event detection and co-reference system with minimal supervision, in the form of a few event examples.

2. Improvement on Entity Co-reference: We improve co-reference resolution on *regular* instances by an ILP-based joint co-reference and mention head detection framework; while on *hard* instances, which require background knowledge, via an algorithmic solution that involves a new representation for knowledge along with a constrained optimization framework.

3. Model Event Sequences via Language Models: We propose two distinct models that capture semantic frame chains and discourse information while abstracting over the specific mentions of predicates and entities.

4. Improvement on Event Sequence Modeling: We augment the developed semantic language models with more semantic aspects, such as sentiments. Moreover, We incorporate background knowledge into our semantic language modeling

1.4 Thesis Overview

This section provides a guide for the rest of the thesis. The thesis can be broadly categorized into four logical segments.

Part I: Background

- Chapter 2

  The first part of this thesis surveys background work. We introduce the relevant NLP tasks, along with the common techniques used for solving them and the evaluation methodologies, i.e. event extraction and co-reference, entity co-reference, discourse parsing and script learning. We then briefly describe the machine learning paradigms and techniques that we apply in this thesis, namely, dataless classification, integer linear programming and neural language models.

Part II: Event Modeling

- Chapter 3

  We first take a close look at the event extraction problem, where we attempt to recognize and categorize events. The biggest challenge is that supervised systems with complex models tend to over-fit on small amounts of training data, and it is expensive to generate a large corpus with complete event annotations. We tackle this problem by developing an event detection system with minimal supervision, in the form of a few event examples. We view the tasks as a semantic similarity problems between event mentions and an ontology of types, thus facilitating the use of large amounts of out of domain text data. Notably, our semantic relatedness function exploits the structure of the text by making use of a semantic-role-labeling based representation of an event. We show that our approach to event detection is competitive with the top supervised methods and support significantly better transfer across domains.

- Chapter 4

  We improve on a previous system for entity co-reference (within document setting) on two fronts: 1) better co-reference decisions on *regular* instances by an ILP-based joint co-reference and mention head detection framework; 2) better co-reference decisions on *hard* instances, which require background knowledge, via an algorithmic solution that involves a new representation for knowledge along with a constrained optimization framework for using this knowledge.

Part III: Event Sequence Modeling

- Chapter 5

  We study a special case of event sequences, i.e. event co-reference resolution. In this thesis, we follow the same semantic similarity formulation for event extraction in Chapter 3 and measure similarities between event mentions in an unsupervised fashion.

- Chapter 6

  We propose that semantic sequences can be modeled as a language model if done at an appropriate level of abstraction. We develop two distinct models that capture semantic frame chains and discourse information while abstracting over the specific mentions of predicates and entities. For each model, we investigate four implementations: a "standard" N-gram language model and three discriminatively trained "neural" language models that generate embeddings for semantic frames. The quality of the semantic language models (SemLM) is evaluated both intrinsically, using perplexity and a narrative cloze test and extrinsically, we show that our SemLM helps improve performance on applications such as co-reference resolution and discourse parsing.

- Chapter 7

  We augment the developed semantic language models with more semantic aspects, such as sentiments. Since not only each individual aspect contributes to modeling a story's semantics, but also the interactions among these aspects, we jointly model important aspects of semantic knowledge – frames, discourse markers, entities and sentiments.

- Chapter 8

  We enhance our modeling of event sequences by incorporating explicit knowledge (beyond the given text). We assume that each event is either generated based on a fact or not. Thus, at each time step, before generating an event, we can predict whether the event to generate corresponds to an underlying fact or not. As a result, the model will provide predictions over facts in addition to predictions over events defined in the semantic language model.

Part IV: Conclusion

- Chapter 9

  The final part of the thesis offers concluding remarks. It summarizes the contributions of this dissertation and identifies directions for future research that can be built on top of this work.

Chapter 2

Background

In this chapter, we first briefly introduce the four main NLP tasks studied in this thesis: event extraction and co-reference, entity co-reference resolution, discourse parsing and script learning. We also discuss the machine learning techniques and paradigms employed in solving these problems: dataless classification, Integer Linear Programming (ILP), and neural language models.

2.1 Related Tasks

In order to give an direct overview of the related tasks in this thesis, we use an example as input (from TAC-KBP 2016 data, and is originally a news piece in New York Times), and illustrate the output of each task in the following subsections.

Task Input Example:
By buying Nokia's handset business, Microsoft may have strengthened its control over the destiny of its mobile operations and gained a potential new chief executive. But completing the $7.2 billion transaction, the technology giant's second-biggest after the acquisition of Skype, was a lengthy process that was anything but straightforward, people briefed on the matter said Tuesday. Steve Ballmer, Microsoft's chief executive, first approached Nokia about a deal during the Mobile World Congress industry conference in Barcelona this year. He emphasized to Stephen Elop, his counterpart at Nokia since 2010, that the software company needed to continue its hardware evolution by developing smaller handsets. Integrating hardware and software, in the mold of Apple, was an important priority. But Microsoft also wanted to ensure that Elop, a former executive, would come along as part of the deal. Nokia at that point felt that Elop was compromised and arranged for Riisto Siilasmaa, the Finnish company's chairman, to take over negotiations. Still, the prospect of Nokia shedding its core business weighed heavily on the company. Nokia's directors met around 50 times in person to discuss virtually every angle of the deal, from valuation to the potential impact on the handset unit's 32,000 workers. Much of the discussions were held directly between Ballmer and Siilasmaa, who met discreetly in Helsinki, London, New York and Seattle, among other cities. The talks moved deliberately, with both sides taking time to figure out how the new structure would work and figure out how to unravel the commercial agreements. This summer, talks between the two sides cooled, as the complexities of the transaction took a toll. They resumed in July, with a broad agreement on the principles of the transaction reached by the end of that month.

2.1.1 Event Extraction and Co-reference

Task Output Example:

Event ID	Text Form	Char Offset	Event Type	Realis Label	Cluster ID
E1	buying	3,9	Transaction.Transaction	Actual	(1)
E2	transaction	199,210	Transaction.Transaction	Actual	(1)
E3	transaction	1698,1709	Transaction.Transaction	Actual	(1)
E4	transaction	1793,1804	Transaction.Transaction	Actual	(1)
E5	acquisition	260,271	Transaction.Transaction	Actual	(2)
E6	said	372,376	Contact.Broadcast	Actual	(3)
E7	met	1151,1154	Contact.Meet	Actual	(4)
...	...	...	...	...	...

Notes: event types are restricted to a pre-defined event type hierarchy in TAC-KBP 2016, and event co-reference output is represented as "Cluster ID".

Event extraction has been studied mainly in the newswire domain as the task of detecting event triggers and determining event types and arguments. Most earlier works have taken a pipeline approach where local classifiers identify triggers first, and then arguments (Ji and Grishman, 2008; Liao and Grishman, 2010; Hong et al., 2011; Huang and Riloff, 2012a,b). Li et al. (2013) presented a structured perceptron model to detect triggers and arguments jointly; while Yang and Mitchell (2016) proposed a method to jointly extract event mentions along with entities. Attempts have also been made to use a Distributional Semantic Model (DSM) to represent events (Gopal and Yang, 2013). A shortcoming of DSMs is that they ignore the structure within the context, thus reducing the distribution to a bag of words. In this thesis, we preserve event structure via structured vector representations constructed from event components. In recent years, as neural networks becoming increasingly popular in NLP, researchers have been employing both recurrent neural networks and convolutional neural networks to extract events (Nguyen et al., 2016; Chang et al., 2016; Feng et al., 2016; Ghaeini et al., 2016; Liu et al., 2017; Tilk et al., 2016; Nguyen and Grishman, 2016). Most of the existing works in event extraction focus on a small of event types, and rely on supervised methods to train and tune on a relatively small dataset (see Chapter 3.2 for details). This often leads to overfitting of the small domain covered in the annotated data. Huang et al. (2016) defined liberal events and came up with an event schema induction framework to generalize the event extraction process. In this thesis, we take a different approach to leverage a better event representation and relieve the reliance on training data; thus making the system transfer better across domains.

Event co-reference is relatively less studied in comparison to the large body of work on entity co-reference. Here, we focus on the within document event co-reference setting. Our work largely follows the event co-reference definition in Hovy et al. (2013). All previous works on event co-reference except Cybulska and Vossen (2012) deals only with full co-reference. Early works (Humphreys et al., 1997; Bagga and Baldwin,

1999) performed event co-reference on scenario specific events. Both Naughton (2009) and Elkhlifi and Faiz (2009) worked on sentence-level co-reference, which is closer to the definition of Danlos and Gaiffe (2003). Pradhan et al. (2007) dealt with both entity and event coreference by taking a three-layer approach. Chen and Ji (2009) proposed a clustering algorithm using a maximum entropy model with a range of features. Bejan and Harabagiu (2010) built a class of nonparametric Bayesian models using a (potentially infinite) number of features to resolve both within and cross document event co-reference. Lee et al. (2012) formed a system with deterministic layers to make co-reference decisions iteratively while jointly resolving entity and event co-reference. Similarly, Lu and Ng (2017) proposed a system to extract events and make event co-reference decisions at the same time, solving the two tasks jointly. More recently, Hovy et al. (2013) presented an unsupervised model to capture semantic relations and co-reference resolution, but they did not show quantitatively how well their system performed in each of these two cases. Huang et al. (2016) also considered the problem of event clustering. They represented event structures based on AMR (Abstract Meaning Representation) and distributional semantics, and further generated event schemas composing event triggers and argument roles. Recently, TAC has organized Event Nugget Detection and Co-reference Evaluations, resulting in fruitful works, some of which contributed to our work (Liu et al., 2015; Mitamura et al., 2015; Hsi et al., 2015; Sammons et al., 2015; Tsai et al., 2016).

2.1.2 Entity Co-reference Resolution

Task Output Example:
By buying [Nokia]_1's handset business, [Microsoft]_2 may have strengthened [its]_2 control over the destiny of [its]_2 mobile operations and gained a potential new chief executive. But completing the $7.2 billion transaction, the technology giant's second-biggest after the acquisition of Skype, was a lengthy process that was anything but straightforward, people briefed on the matter said Tuesday. [Steve Ballmer]_3, [Microsoft's chief executive]_3, first approached [Nokia]_1 about a deal during the Mobile World Congress industry conference in [Barcelona]_4 this year. [He]_3 emphasized to [Stephen Elop]_5, [his]_3 counterpart at [Nokia]_1 since 2010, that the [software company]_1 needed to continue [its]_1 hardware evolution by developing smaller handsets. .....
Notes: mentions are in brackets, and the index indicates entity clusters.

Co-reference resolution determines whether two entities/events refer to the same thing. Here, we focus on the within document entity co-reference problem. The task has been extensively studied, with several state-of-the-art approaches addressing this task (Lee et al., 2011; Song et al., 2012; Durrett and Klein, 2013; Björkelund and Kuhn, 2014). Many of the early rule-based systems like Hobbs (1978) and Lappin and Leass (1994) gained considerable popularity. The early designs were easy to understand and the rules were designed manually. Machine learning approaches were introduced in many works (Connolly et al., 1997;

Ng and Cardie, 2002b; Bengtson and Roth, 2008; Soon et al., 2001). The introduction of ILP methods has influenced the coreference area too (Chang et al., 2011; Denis and Baldridge, 2007). Recent neural network based methods achieve the current best results on benchmark evaluation datasets (Clark and Manning, 2015; Wiseman et al., 2015, 2016; Clark and Manning, 2016b,a; **?**; Peters et al., 2018).

A pre-requisite step for entity co-reference is mention detection. It is closely related to *Named Entity Recognition* (NER). Punyakanok and Roth (2001) thoroughly study phrase identification in sentences and propose three different general approaches. They aim to learn several different local classifiers and combine them to optimally satisfy some global constraints. Cardie and Pierce (1998) propose to select certain rules based on a given corpus, to identify base noun phrases. However, the phrases detected are not necessarily mentions that we need to discover. Ratinov and Roth (2009) present detailed studies on the task of named entity recognition, which discusses and compares different methods on multiple aspects including chunk representation, inference method, utility of non-local features, and integration of external knowledge. NER can be regarded as a sequential labeling problem, which can be modeled by several proposed models, e.g. Hidden Markov Model (Rabiner, 1989) or Conditional Random Fields (Sarawagi and Cohen, 2004). The typical BIO representation was introduced in Ramshaw and Marcus (1995); OC representations were introduced in Church (1988), while Finkel and Manning (2009) further study nested named entity recognition, which employs a tree structure as a representation of identifying named entities within other named entities.

The most relevant study on mentions in the context of coreference was done in Recasens et al. (2013); this work studies distinguishing single mentions from coreferent mentions. The proposed joint framework (Chapter 4.1) provides similar insights, where the added mention decision variable partly reflects if the mention is singleton or not. Several recent works suggest studying coreference jointly with other tasks. Lee et al. (2012) model entity coreference and event coreference jointly; Durrett and Klein (2014) consider joint coreference and entity-linking. Lassalle and Denis (2015) propose a joint anaphoricity detection and coreference resolution framework while assuming gold mentions are given.

2.1.3 Shallow Discourse Parsing

Task Output Example:
[By buying Nokia's handset business, Microsoft may have strengthened its control over the destiny of its mobile operations and gained a potential new chief executive.]_*Arg1* <u>But</u> [completing the $7.2 billion transaction, the technology giant's second-biggest after the acquisition of Skype, was a lengthy process that was anything but straightforward]_*Arg2*, people briefed on the matter said Tuesday. .....
Notes: the identified connective is underlined, and we also output the text span of Arg1 and Arg2.

Shallow discourse parsing is the task of identifying explicit and implicit discourse connectives, determine

their senses and their discourse arguments. There has been a surge of interest in shallow discourse parsing since the release of PDTB (Prasad et al., 2008), the first large discourse corpus distinguishing implicit examples from explicit ones. A large set of work has focused on direct classification based on observed sentences, including structured methods with linguistically-informed features (Lin et al., 2009; Pitler et al., 2009; Zhou et al., 2010), end-to-end neural models (Qin et al., 2016a,b; Chen et al., 2016; Liu and Li, 2016), and combined approaches (Ji and Eisenstein, 2015; Ji et al., 2016).

The most difficult sub-problem for discourse parsing is to identify implicit discourse connectives and the corresponding senses. The lacking of connective cues makes learning purely from contextual semantics full of challenges. Prior work has attempted to leverage connective information (Zhou et al., 2010). Another notable line of work aims at adapting explicit examples for data synthesis (Biran and McKeown, 2013; Rutherford and Xue, 2015; Braud and Denis, 2015; Ji and Eisenstein, 2015), multi-task learning (Lan et al., 2013; Liu et al., 2016), and word representation (Braud and Denis, 2016).

In this thesis, we employ shallow discourse parsing for two purposes (see Chapter 6 for details):

- Explicit connective extraction: we use an existing shallow discourse parsing system (Song et al., 2015) to extract explicit connectives as a pre-processing step.
- Application task: in order to show the effectiveness of the proposed event sequence modeling technique, we use shallow discourse parsing as the application task, where we follow the setting of CoNLL-2016 Shared Task (Xue et al., 2016).

2.1.4 Script Learning

Event sequence modeling is related to script learning. Early works (Schank and Abelson, 1977; Mooney and DeJong, 1985) tried to construct knowledge bases from documents to learn scripts. Recent work focused on utilizing statistical models to extract high-quality scripts from large amounts of data (Chambers and Jurafsky, 2008; Bejan, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Granroth-Wilding et al., 2015; Pichotta and Mooney, 2016). Other works aimed at learning a collection of structured events (Chambers, 2013; Cheung et al., 2013; Balasubramanian et al., 2013; Bamman and Smith, 2014; Nguyen et al., 2015), and several works have employed neural embeddings (Modi and Titov, 2014; Frermann et al., 2014; Titov and Khoddam, 2015). Ferraro and Van Durme (2016) presented a unified probabilistic model of syntactic and semantic frames while also demonstrating improved coherence.

However, previous research has the following limitations: 1) script learning does not generate a probabilistic model on semantic frames[1]; 2) script learning models semantic frame sequences incompletely as they do

[1]Some works may utilize a certain probabilistic framework, but they mainly focus on generating high-quality frames by

not consider discourse information; 3) works in script learning rarely show applications to real NLP tasks. Some prior works have used scripts-related ideas to help improve NLP tasks (Irwin et al., 2011; Rahman and Ng, 2011). However, since they use explicit script schemas either as features or constraints, these works suffer from data sparsity problems. In this thesis, we address this problem by abstracting over various semantic units, including semantic frames, entities, discourse markers, sentiments, etc.

Most recently, Mostafazadeh et al. (2016, 2017) proposed story cloze test as a standard way to test a system's ability to model semantics. They released ROCStories dataset, and organized a shared task for LSDSem'17 (Mostafazadeh et al., 2017); which yields many insightful works on this task. Cai et al. (2017) developed a model that uses hierarchical recurrent networks with attention to encode sentences and produced a strong baseline.

2.2 Related Techniques

2.2.1 Dataless Classification

Dataless classification performs a nearest neighbor search of labels for a piece of text in an appropriately selected semantic space (Chang et al., 2008; Song and Roth, 2014). The core problem in dataless classification is to find a semantic space that enables good representations of texts and labels. Traditional text classification makes use of a bag-of-words (BOW) representation of documents. However, when comparing labels and texts in dataless classification, the brevity of labels makes this simple minded representation and the resulting similarity measure unreliable. For example, a document talking about "sports" does not necessarily contain the word "sports." Consequently, other more expressive distributional representations have been applied, e.g., Brown cluster (Brown et al., 1992; Liang, 2005), neural network embedding (Turian et al., 2010; Collobert et al., 2011; Mikolov et al., 2013a), topic modeling (Blei et al., 2003), ESA (Gabrilovich and Markovitch, 2009), and their combinations (Song and Roth, 2015). It has been shown that ESA gives the best and most robust results for dataless classification for English documents (Song and Roth, 2014).

In this thesis, we follow the dataless classification learning paradigm for both event extraction and event co-reference. Note that it requires even less supervision than unsupervised methods, which normally operates on a large in-domain corpus. More details on the comparison between dataless classification and supervised/unsupervised methods are provided in Chapter 3.1.

filtering.

2.2.2 Integer Linear Program

An integer linear program in canonical form is expressed as:

$$\begin{aligned} \arg\max_{\mathbf{y}} \quad & \mathbf{w}^\top \mathbf{y} \\ \text{s.t.} \quad & A\mathbf{y} \leq b \\ & \mathbf{y} \geq 0, \quad \mathbf{y} \in \mathcal{Z}^n. \end{aligned}$$

It is a constrained optimization problem, where decision variables y are constrained to be integers. The ILP problem is proven to be NP-Hard (Schrijver, 1998).

To solve ILPs, there are a variety of algorithms that can be used to get exact solutions. One class of algorithms are cutting plane methods (Marchand et al., 2002) which work by solving the LP relaxation and then adding linear constraints that drive the solution towards being integer without excluding any integer feasible points. Another class of algorithms are variants of the branch and bound method (Land and Doig, 1960). For example, the branch and cut method that combines both branch and bound and cutting plane methods. Branch and bound algorithms have a number of advantages over algorithms that only use cutting planes. One advantage is that the algorithms can be terminated early and as long as at least one integral solution has been found, a feasible, although not necessarily optimal, solution can be returned. Further, the solutions of the LP relaxations can be used to provide a worst-case estimate of how far from optimality the returned solution is. Finally, branch and bound methods can be used to return multiple optimal solutions.

In this thesis, we apply ILP formulation for the coreference problem in particular. Thus, y becomes the binary decision variable to determine whether or not two mentions should be linked; while the corresponding w indicates the mention pair score (which is to be learned). Depending on different coreference inference protocols (Chang et al., 2011), we add different constraints in this ILP formulation.

2.2.3 Neural Language Models

Neural models can learn word representations that aid in making predictions within local context windows. For example, Bengio et al. (2003) introduced a model that learns word vector representations as part of a simple neural network architecture for language modeling. Collobert and Weston (2008) decoupled the word vector training from the downstream training objectives, which paved the way for Collobert et al. (2011) to use the full context of a word for learning the word representations, rather than just the preceding context as is the case with language models. Recently, the importance of the full neural network structure for learning useful word representations has been called into question. The skip-gram and continuous bag-

of-words (CBOW) models of Mikolov et al. (2013a) propose a simple single-layer architecture based on the inner product between two word vectors. Mnih and Kavukcuoglu (2013) also proposed closely-related vector log-bilinear models, vLBL and ivLBL, and Levy and Goldberg (2014b) proposed explicit word embeddings based on a PPMI metric. In the skip-gram and ivLBL models, the objective is to predict a word's context given the word itself, whereas the objective in the CBOW and vLBL models is to predict a word given its context. Through evaluation on a word analogy task, these models demonstrated the capacity to learn linguistic patterns as linear relationships between the word vectors.

In this dissertation, we model event sequences via neural language models. Instead of modeling word tokens, language models operate on semantic units, such as semantic frames, abstracted entities, extracted discourse markers, etc. Like all language model evaluations, we also report perplexity numbers for the developed semantic language models as an intrinsic quality check. We provide more details in Chapter 6.

Chapter 3

Event Extraction

Natural language understanding involves, as a key component, the need to understand events mentioned in texts. This entails recognizing elements such as agents, patients, actions, location and time, among others. Events have been studied for years, but they still remain a key challenge. One reason is that the frame-based structure of events necessitates addressing multiple coupled problems that are not easy to study in isolation. Perhaps an even more fundamental difficulty is that it is not clear whether our current set of events' definitions is adequate (Hovy et al., 2013). Thus, given the complexity and fundamental difficulties, the current evaluation methodology in this area focuses on a limited domain of events, e.g. 33 types in ACE 2005 (NIST, 2005) and 38 types in TAC KBP (Mitamura et al., 2015). Consequently, this allows researchers to train supervised systems that are tailored to these sets of events and that overfit the small domain covered in the annotated data, rather than address the realistic problem of understanding events in text. In this chapter, we pursue an approach to extract events that we believe to be more feasible and scalable.

3.1 Overview of Minimally Supervised Event Pipeline

We pursue an approach to understanding events that we believe to be more feasible and scalable. Fundamentally, event detection is about identifying whether an event in context is semantically related to a set of events of a specific type. Therefore, if we formulate event detection and co-reference as semantic relatedness problems, we can scale it to deal with a lot more types and, potentially, generalize across domains. Moreover, by doing so, we facilitate the use of a lot of data that is not part of the existing annotated event collections and not even from the same domain. The key challenges we need to address are those of how to represent events, and how to model event similarity; both are difficult partly since events have *structure*.

We present a general event detection and co-reference framework[1], which essentially requires no labeled data. In practice, in order to map an event mention to an event ontology, as a way to communicate with a user, we just need a few event examples, in plain text, for each type a user wants to extract. This is

[1]The event co-reference details will be discussed in Chapter 4.3

	Supervised	Unsupervised	MSEP
Guideline	✓	✓	✓
In-domain Data	✓	✓	✗
Data Annotation	✓	✗	✗

Table 3.1: Comparing requirements of MSEP and other methods. Supervised methods need all three resources while MSEP only needs an annotation guideline (as event examples).

a reasonable setting; after all, giving examples is the easiest way of defining event types, and is also how information needs are defined to annotators - by providing examples in the annotation guideline.[2] Our approach makes less assumptions than standard *unsupervised* methods, which typically require a *collection* of instances and exploit similarities among them to eventually learn a model. Here, given event type definitions (in the form of a few examples), we can classify a *single* event into a provided ontology and determine whether two events are co-referent. In this sense, our approach is similar to what has been called *dataless classification* (Chang et al., 2008; Song and Roth, 2014). Table 3.1 summarizes the difference between our approach, MSEP (Minimally Supervised Event Pipeline), and other methods.

Our approach builds on two key ideas. First, to represent event structures, we use the general purpose nominal and verbial semantic role labeling (SRL) representation. This allows us to develop a structured representation of an event. Second, we embed event components, while maintaining the structure, into multiple semantic spaces, induced at a contextual, topical, and syntactic levels. These semantic representations are induced from large amounts of text in a way that is completely independent of the tasks at hand, and are used to represent both event mentions and event types into which we classify our events. The combination of these semantic spaces, along with the structured vector representation of an event, allow us to directly determine whether a candidate event mention is a valid event or not and, if it is, of which type. Moreover, with the same representation, we can evaluate event similarities and decide whether two event mentions are co-referent. Consequently, the proposed MSEP, can also adapt to new domains without any training. An overview of the system is shown in Figure fig:flow. A few event examples are *all* the supervision MSEP needs; even the few decision thresholds needed to be set are determined on these examples, once and for all, and are used for *all* test cases we evaluate on.

[2]Event examples also serve for disambiguation purposes. For example, using "U.S. forces bombed Baghdad." to exemplify an *attack* type, disambiguates it from a *heart attack*.

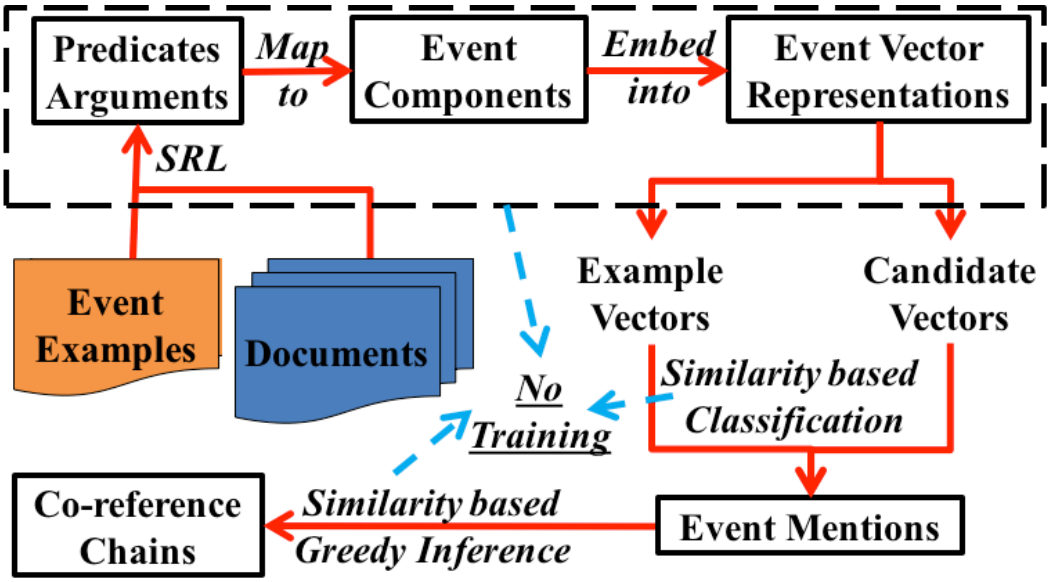

Figure 3.1: An overview of the end-to-end MSEP system. "Event Examples" are the only supervision. No training is needed for MSEP. Event co-reference will be discussed in Chapter 5.

3.2 Structured Vector Representation

There is a parallel between event structures and sentence structures. Event triggers are mostly predicates of sentences or clauses. Event arguments are largely entity mentions or temporal/spatial arguments. They serve as specific roles in events, similarly to SRL arguments that are assigned role labels for predicates. Thus, we identify the five most important and abstract event semantic components – action, agent$_{sub}$, agent$_{obj}$, location and time by mapping from SRL arguments. Details can be referred to in Peng et al. (2016).

We use the Illinois SRL (Punyakanok et al., 2004) tool to pre-process the text. We evaluate the SRL coverage on both event triggers and event arguments, shown in Table 3.2.[3] For event triggers, we only focus on recall since we expect the event mention detection module to filter out most non-trigger predicates. Results show a good coverage of SRL predicates and arguments on event triggers and arguments. Even though we only get approximate event arguments, it is easier and more reliable to categorize them into five abstract roles, than to determine the exact role label with respect to event triggers.

To map SRL arguments to these event arguments, we run through the following procedures.

1. Set predicates as actions, and preserve SRL negations for actions.

2. Set SRL subject as agent$_{sub}$.

[3]We place events in two categories, verb or noun, according to the part-of-speech tag of the trigger. We evaluate verb-SRL on events with verb triggers, nom-SRL on events with noun triggers, and the overall performance on all events. When evaluating, we allow partial overlaps.

3. Set SRL object and indirect object as agent_{obj}

4. Set SRL spatial argument as event location. If there is no such SRL label, we then scan for any NER location label within the sentence/clause to which the action belongs. We set the location according to NER information if it exists.

5. Set the SRL temporal argument as event time. If there is no such SRL label, we then use the Illinois Temporal Expression Extractor (Zhao et al., 2012) to find the temporal argument within an event's sentence/clause.

We allow one or more missing event arguments among agent_{sub}, agent_{obj}, location or time, but require actions to always exist.

Given the structured information, we convert each event component to its corresponding vector representation. We then concatenate the vectors of all components together in a specific order: action, agent_{sub}, agent_{obj}, location, time and sentence/clause. We treat the whole sentence/clause, to which the "action" belongs, as context, and we append its corresponding vector to the event representation. This basic event vector representation is illustrated in Fig. 3.2. If there are missing event arguments, we set the corresponding vector to be "NIL" (we set each position as "NaN"). We also augment the event vector representation by concatenating more text fragments to enhance the interactions between the action and other arguments, as shown in Fig. 3.3. Essentially, we flatten the event structure to preserve the alignment of event arguments so that the structured information can be reflected in our vector space.

ACE		Precision	Recall	F1
Predicates	Verb-SRL	—	93.2	—
over	Nom-SRL	—	87.5	—
Triggers	All	—	91.9	—
SRL Args	Verb-SRL	90.4	85.7	88.0
over	Nom-SRL	92.5	73.5	81.9
Event Args	All	90.9	82.3	86.4
TAC KBP		Precision	Recall	F1
Predicates	Verb-SRL	—	90.6	—
over	Nom-SRL	—	85.5	—
Triggers	All	—	88.1	—
SRL Args	Verb-SRL	89.8	83.6	86.6
over	Nom-SRL	88.2	69.9	78.0
Event Args	All	89.5	81.0	85.0

Table 3.2: Semantic role labeling coverage. We evaluate both "Predicates over Triggers" and "SRL Arguments over Event Arguments". "All" stands for the combination of Verb-SRL and Nom-SRL. The evaluation is done on all data.

3.3 Text-Vector Conversion

We experiment with different methods to convert event components into vector representations. Specifically, we use Explicit Semantic Analysis (ESA), Brown Cluster (BC), Word2Vec (W2V) and Dependency-Based Word Embedding (DEP) respectively to convert text into vectors. We then concatenate all components of an event together to form a structured vector representation.

- Explicit Semantic Analysis

  ESA uses Wikipedia as an external knowledge base to generate concepts for a given fragment of text (Gabrilovich and Markovitch, 2009). ESA first represents a given text fragment as a TF-IDF vector, then uses an inverted index for each word to search the Wikipedia corpus. The text fragment representation is thus a weighted combination of the concept vectors corresponding to its words. We use the same setting as in Chang et al. (2008) to filter out pages with fewer than 100 words and those containing fewer than 5 hyper-links. To balance between the effectiveness of ESA representations and its cost, we use the 200 concepts with the highest weights. Thus, we convert each text fragment to a very sparse vector of millions of dimensions (but we just store 200 non-zero values). Note that the extracted 200 concepts are not the same for all words. When we compute the similarity of two ESA vectors, we only consider the similarity between their shared concepts.

- Brown Cluster

  BC was proposed by Brown et al. (1992) as a way to support abstraction in NLP tasks, measuring words' distributional similarities. This method generates a hierarchical tree of word clusters by evaluating the word co-occurrence based on a n-gram model. Then, paths traced from root to leaves can be used as word representations. We use the implementation by Song and Roth (2014), generated over the latest Wikipedia dump. We set the maximum tree depth to 20, and use a combination of path prefixes of length 4,6 and 10 as our BC representation. Thus, we convert each word to a vector of $2^4 + 2^6 + 2^{10} = 1104$ dimensions.

- Word2Vec

  We use the skip-gram tool by Mikolov et al. (2013c) over the latest Wikipedia dump, resulting in word vectors of dimensionality 200.

- Dependency-Based Embedding

  DEP is the generalization of the skip-gram model with negative sampling to include arbitrary contexts. In particular, it deals with dependency-based contexts, and produces markedly different embeddings.

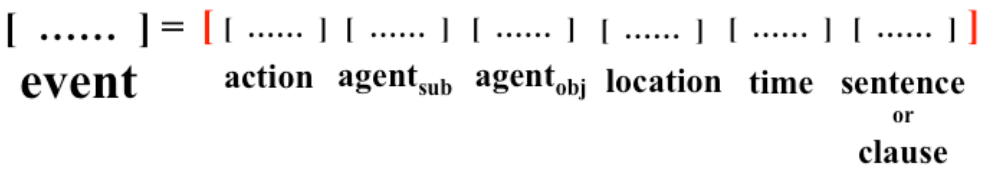

Figure 3.2: Basic event vector representation. Event vector is the concatenation of vectors corresponding to action, agent$_{sub}$, agent$_{obj}$, location, time and sentence/clause.

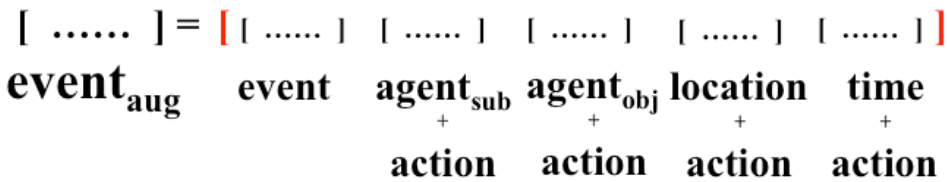

Figure 3.3: Augmented event vector representation. Event vector is the concatenation of vectors corresponding to basic event vector representation, agent$_{sub}$ + action, agent$_{obj}$ + action, location + action and time + action. Here, "+" means that we first put text fragments together and then convert the combined text fragment into an ESA vector.

DEP exhibits more functional similarity than the original skip-gram embeddings (Levy and Goldberg, 2014a). We directly use the released 300-dimension word embeddings.

Note that it is straightforward text-vector conversion for ESA. But for BC, W2V and DEP, we first remove stop words from the text and then average, element-wise, all remaining word vectors to produce the resulting vector representation of the text fragment.

3.4 Event Mention Detection

Motivated by the seed-based event trigger labeling technique employed in Bronstein et al. (2015), we turn to ACE annotation guidelines for event examples described under each event type label. For instance, the ACE-2005 guidelines [4] list the example "Mary Smith joined Foo Corp. in June 1998." for label "START-POSITION". Altogether, we collect 172 event examples from 33 event types (see Appendix for the complete list). We define the *event type representation* as the numerical average of all vector representations corresponding to example events under that type. We use the similarity between an event candidate with the

[4] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

event type representation to determine whether the candidate belongs to an event type:

$$
\begin{aligned}
S(e_1, e_2) &= \frac{vec(e_1) \cdot vec(e_2)}{\|vec(e_1)\| \|vec(e_2)\|} \\
&= \frac{\sum_{\text{arg}} vec(e_1^{\text{arg}}) \cdot vec(e_2^{\text{arg}})}{\sqrt{\sum_{\text{arg}} \|vec(e_1^{\text{arg}})\|^2} \sqrt{\sum_{\text{arg}} \|vec(e_2^{\text{arg}})\|^2}},
\end{aligned} \tag{3.1}
$$

where e_1 is the candidate, e_2 the type ($vec(e_2)$ is computed as average of event examples), $e_1^{\text{arg}}, e_2^{\text{arg}}$ are components of e_1, e_2 respectively. We use the notation $vec(\cdot)$ for corresponding vectors. Note that there may be missing event arguments (NIL). In such cases, we use the average of all non-NIL similarity scores for that particular component as the contributed score. Formally, we define $S_{pair}(a = \text{NIL})$ and $S_{single}(a = \text{NIL})$ as follows:

$$
\begin{aligned}
S_{pair}(e^{\text{arg}} = \text{NIL}) &= vec(\text{NIL}) \cdot vec(e_2^{\text{arg}}) \\
&= vec(e_1^{\text{arg}}) \cdot vec(\text{NIL}) \\
&= \sum_{e_1^{\text{arg}}, e_2^{\text{arg}} \neq \text{NIL}} \frac{vec(e_1^{\text{arg}}) \cdot vec(e_2^{\text{arg}})}{\#|e_1^{\text{arg}}, e_2^{\text{arg}} \neq \text{NIL}|},
\end{aligned}
$$

$$
S_{single}(e^{\text{arg}} = \text{NIL}) = \sqrt{\frac{\sum_{e^{\text{arg}} \neq \text{NIL}} \|vec(e^{\text{arg}})\|^2}{\#|e^{\text{arg}} \neq \text{NIL}|}}.
$$

Thus, when we encounter missing event arguments, we use $S_{pair}(e^{\text{arg}} = \text{NIL})$ to replace the corresponding term in the numerator in $S(e_1, e_2)$ while using $S_{single}(e^{\text{arg}} = \text{NIL})$ in the denominator. These average contributed scores are corpus independent, and can be pre-computed ahead of time. We use a cut-off threshold to determine that an event does not belong to any event types, and can thus be eliminated. This threshold is set by tuning only on the set of event examples, which is corpus independent.

3.5 Experiments

We use two benchmark datasets, i.e. ACE-2005 and TAC-KBP (Statistics shown in Table 3.3.) to compare MSEP with baselines and supervised systems. The superiority of MSEP is also demonstrated in across domain settings.

- ACE Dataset

  The ACE-2005 English corpus (NIST, 2005) contains fine-grained event annotations, including event trigger, argument, entity, and time-stamp annotations. We select 40 documents from newswire articles for event detection evaluation and the rest for training (same as Chen et al. (2015)).

	#Doc	#Sent.	#Men.	#Cluster
ACE(All)	599	15,494	5,268	4,046
ACE(Test)	40	672	289	222
TAC-KBP(All)	360	15,824	12,976	7,415
TAC-KBP(Test)	202	8,851	6,438	3,779

Table 3.3: Statistics for the ACE and TAC-KBP corpora. #Sent. is the number of sentences, #Men. is the number of event mentions, and #Cluster is the number of event clusters (including singletons). Note that the proposed MSEP does not need any training data. ACE(Test) is only used to evaluate event detection while we do cross-validation for ACE event co-reference. TAC-KBP(Test) is used for both event detection and co-reference evaluations.

Data	Method	Span			Span+Type		
		Precision	Recall	F1	Precision	Recall	F1
ACE	DMCNN	80.4	67.7	73.5	75.6	63.6	**69.1**
	SSED	76.6	71.5	**74.0**	71.3	66.5	68.8
	MSEP-EMD	75.6	69.8	72.6	70.4	65.0	67.6
TAC-KBP	SSED	77.2	55.9	64.8	69.9	48.8	57.5
	TAC-TOP	—	—	**65.3**	—	—	**58.4**
	MSEP-EMD	76.5	54.5	63.5	69.2	47.8	56.6

Table 3.4: Event Extraction (trigger identification) results.

- TAC-KBP Dataset

  The TAC-KBP-2015 corpus is annotated with event nuggets that fall into 38 types and co-reference relations between events. [5] We use the train/test data split provided by the official TAC-2015 Event Nugget Evaluation Task. Note that the training set and cross-validation is only for competing supervised methods. For MSEP, we only need to run on each corpus once for testing.

- Compared Systems

  For event detection, we compare with **DMCNN** (Chen et al., 2015), the state-of-art supervised event detection system. We also implement another supervised model, named *supervised structured event detection* **SSED** system following the work of Sammons et al. (2015). The system utilizes rich semantic features and applies a trigger identification classifier on every SRL predicate to determine the event type. We name our event mention detection module in MSEP *similarity-based event mention detection* **MSEP-EMD** system.

- Evaluation Metrics

  For event detection, we use standard precision, recall and F1 metrics.

[5]The event ontology of TAC-KBP (based on ERE annotation) is almost the same to that of ACE. To adapt our system to the TAC-KBP corpus, we use all ACE event seeds of "Contact.Phone-Write" for "Contact.Correspondence" and separate ACE event seeds of "Movement.Transport" into "Movement.TransportPerson" and "Movement.TransportArtifact" by manual checking. So, we use exactly the same set of event seeds for TAC-KBP with only these two changes.

	Train	Test	MSEP	Supervised
Event Detection			Span+Type F1	
In Domain	NW	NW	58.5	**63.7**
Out of Domain	DF	NW	**55.1**	54.8
In Domain	DF	DF	57.9	**62.6**
Out of Domain	NW	DF	**52.8**	52.3

Table 3.5: Domain Transfer Results. We conduct the evaluation on TAC-KBP corpus with the split of newswire (NW) and discussion form (DF) documents. Here, we choose MSEP-EMD for event detection. We use SSED as the supervised module for comparison. We compare F1 scores of span plus type match.

Results for Event Extraction

The performance comparison for event extraction is presented in Table 3.4. On both ACE and TAC-KBP, parameters of SSED are tuned on a development set (20% of randomly sampled training documents). The cut-off threshold for MSEP-EMD is tuned on the 172 event examples ahead of time by optimizing the F1 score on the event seed examples. Note that different text-vector conversion methods lead to different cut-off thresholds, but they remain fixed for all the test corpus. Results show that SSED achieves state-of-the-art performance. Though MSEP-EMD's performance is below the best supervised system, it is very competitive. Note that both SSED and MSEP-EMD use SRL predicates as input and thus can further improve with a better SRL module. There is a notable performance drop on TAC-KBP, compared to that on ACE. One likely reason is that events are much denser in TAC-KBP than in ACE. On average, there are 0.82 events in a sentence for TAC-KBP while it is only 0.34 for ACE.

Domain Transfer Evaluation

To demonstrate the superiority of the adaptation capabilities of the proposed MSEP system, we test its performance on new domains and compare with the supervised system. TAC-KBP corpus contains two genres: newswire (NW) and discussion forum (DF), and they have roughly equal number of documents. When trained on NW and tested on DF, supervised methods encounter out-of-domain situations. However, the MSEP system can adapt well.[6] Table 3.5 shows that MSEP outperforms supervised methods in out-of-domain situations.

[6]Note that the supervised method needs to be re-trained and its parameters re-tuned while MSEP does not need training and its cut-off threshold is fixed ahead of time using event examples.

Chapter 4

Entity Coreference Resolution

Coreference resolution is a key problem in natural language understanding that still escapes reliable solutions. In this chapter, we present works on improving entity coreference for entities. We propose a joint coreference resolution and mention head detection framework to improve the performance on predicted mentions; and also improve on hard instances by incorporating external knowledge as constraints during inference.

4.1 Joint Framework for Mention Head Detection and Co-reference

4.1.1 Motivation

In coreference resolution, a fair amount of research treats mention detection as a pre-processed step and focuses on developing algorithms for clustering coreferred mentions. However, there are significant gaps between the performance on gold mentions and the performance on the real problem, when mentions are *predicted* from raw text via an imperfect Mention Detection (MD) module (as shown in Table 4.1). Motivated by the goal of reducing such gaps, we develop an ILP-based joint coreference resolution and mention head formulation that is shown to yield significant improvements on coreference from raw text, outperforming existing state-of-art systems on both the ACE-2004 and the CoNLL-2012 datasets (as in June 2015). At the same time, our joint approach is shown to improve mention detection by close to 15% F1. One key insight underlying our approach is that identifying and co-referring mention *heads* is not only sufficient but is more robust than complete mentions.

4.1.2 System Design

In this work, we focus on improving end-to-end coreference performance. We do this by:

1. Developing a new ILP-based joint learning and inference formulation for coreference and mention head detection.

System	Dataset	Gold	Predict	Gap
Illinois (Chang et al., 2013)	CoNLL-12	77.05	60.00	17.05
Illinois (Chang et al., 2013)	CoNLL-11	77.22	60.18	17.04
Illinois (Chang et al., 2013)	ACE-04	79.42	68.27	11.15
Berkeley (Durrett and Klein, 2013)	CoNLL-11	76.68	60.42	16.26
Stanford (Lee et al., 2011)	ACE-04	81.05	70.33	10.72

Table 4.1: Performance gaps between using gold mentions and predicted mentions. Performance gaps are always larger than 10%. Illinois's system (Chang et al., 2013) is evaluated on CoNLL (2012, 2011) Shared Task and ACE-2004 datasets. It reports an average F1 score of MUC, B^3 and $CEAF_e$ metrics using CoNLL v7.0 scorer. Berkeley's system (Durrett and Klein, 2013) reports the same average score on the CoNLL-2011 Shared Task dataset. Results of Stanford's system (Lee et al., 2011) are for B^3 metric on ACE-2004 dataset.

2. Developing a better mention head candidate generation algorithm.

Importantly, we focus on heads rather than mention boundaries since those can be identified more robustly and used effectively in an end-to-end system. As we show, this results in a dramatic improvement in the quality of the MD component and, consequently, a significant reduction in the performance gap between coreference on gold mentions and coreference on raw data.

We develop a joint coreference resolution and mention head detection framework as an Integer Linear Program (ILP) following Roth and Yih (2004). Figure 4.1 compares a traditional pipelined system with our proposed system. Our joint formulation includes decision variables both for coreference links between pairs of mention heads, and for all mention head candidates, and we simultaneously learn the ILP coefficients for all these variables. During joint inference, some of the mention head candidates will be rejected (that is, the corresponding variables will be assigned '0'), contributing to improvement both in MD and in coreference performance. The aforementioned joint approach builds on an algorithm that generates mention head candidates. Our candidate generation process consists of a statistical component and a component that makes use of existing resources, and is designed to ensure high recall on head candidates.

Existing coreference systems usually consider a pipelined system, where the mention detection step is followed by that of clustering mentions into coreference chains. Higher quality mention identification naturally leads to better coreference performance. Standard methods define mentions as *boundaries* of text, and expect *exact* boundaries as input in the coreference step. However, mentions have an intrinsic structure, in which mention heads carry the crucial information. Here, we define a mention head as the last token of a syntactic head, or the whole syntactic head for proper names.[1] For example, in "the incumbent [Barack Obama]" and "[officials] at the Pentagon", "Barack Obama" and "officials" serve as mention heads, respectively. Mention heads can be used as auxiliary structures for coreference. In this work, we first identify mention heads, and then detect mention boundaries based on heads. We rely heavily on the first, head identification, step,

[1]Here, we follow the ACE annotation guideline. Note that the CoNLL-2012 dataset is built from OntoNotes-5.0 corpus.

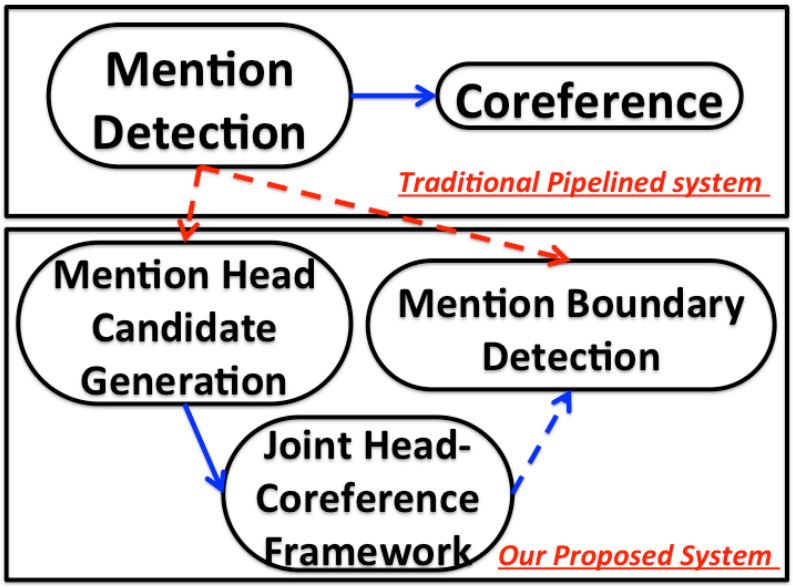

Figure 4.1: Comparison between a traditional pipelined system and our proposed system. We split up mention detection into two steps: mention head candidate generation and (an optional) mention boundary detection. We feed mention heads rather than complete mentions into the coreference model. During the joint head-coreference process, we reject some mention head candidates and then recover complete mention boundaries after coreference decisions are made.

which we show to be sufficient to support coreference decisions. Moreover, this step also provides enough information for "understanding" the coreference output, and can be evaluated more robustly (since minor disagreements on mention boundaries are often a reason for evaluation issues when dealing with predicted mentions). We only identify the mention boundaries at the end, after we make the coreference decisions, to be consistent with current evaluation standards in the coreference resolution community. Consider the following example[2]:

> [Multinational companies investing in [China]] had become so angry that [they] recently set up an anti-piracy *league* to pressure [the [Chinese] government] to take action. [Domestic manufacturers, [who] are also suffering], launched a similar body this month. [They] hope [the government] can introduce a new law increasing fines against [producers of fake goods] from the amount of profit made to the value of the goods produced.

Here, phrases in the brackets are mentions and the underlined simple phrases are mention heads. Moreover, mention boundaries can be nested (the boundary of a mention is inside the boundary of another mention), but mention heads never overlap. This property also simplifies the problem of mention head candidate generation. In the example above, the first "they" refers to "Multinational companies investing in China" and the second "They" refers to "Domestic manufacturers, who are also suffering". In both cases, the

[2]This example is chosen from the ACE-2004 corpus.

mention heads are sufficient to support the decisions: "they" refers to "companies", and "They" refers to "manufacturers". In fact, most of the features[3] implemented in existing coreference resolution systems rely solely on mention heads (Bengtson and Roth, 2008). Furthermore, consider the possible mention candidate "league" (italic in the text). It is not chosen as a mention because the surrounding context is not focused on "anti-piracy league". Mention detection can also be viewed as a global decision problem, which involves considering the relevance of a mention to its context. The fact that the coreference decision provides a way to represent this relevance, further motivates considering mention detection and coreference jointly. The insight here is that a mention candidate will be more likely to be valid when it has more high confidence coreference links. Ideally, after making coreference decisions, we extend the remaining mention heads to complete mentions; we employ a binary classifier, which shares all features with the mention head detection model in the joint step.

Our proposed system can work on both ACE and OntoNotes datasets, even though their styles of annotation are different. There are two main differences to be addressed. First, OntoNotes removes singleton mentions, even if they are valid mentions. This causes additional difficulty in learning a good mention detector in a pipelined framework. However, our joint framework can adapt to it by rejecting those singleton mentions. More details will be discussed in Sec. 2. Second, ACE uses shortest denotative phrases to identify mentions while OntoNotes tends to use long text spans. This makes identifying mention boundaries unnecessarily hard. Our system focuses on mention heads in the coreference stage to ensure robustness. As OntoNotes does not contain head annotations, we preprocess the data to extract mention heads which conform with the ACE style.

The main contributions of this work can be summarized as follows:

1. We develop a new, end-to-end, coreference approach that is based on a joint learning and inference model for mention heads and coreference decisions.

2. We develop an improved mention head candidate generation module and a mention boundary detection module.

3. We achieve the best coreference results on predicted mentions and reduce the performance gap compared to using gold mentions.

[3] All features except for those that rely on modifiers.

4.1.3 Mention Head Candidate Generation

The goal of the mention head candidate generation process is to acquire candidates from multiple sources to ensure high recall, given that our joint framework acts as a filter and increases precision. We view the sources as independent components and merge all mention heads generated.

- Sequence Labeling Component

  The sequence labeling component builds on the assumption that different mentions have different heads, and heads do not overlap with each other. The problem of identifying mention heads is thus a sequential phrase identification problem, and we choose to employ the $BILOU$-representation as it has advantages over traditional BIO-representation, as shown, e.g. in Ratinov and Roth (2009). The $BILOU$-representation suggests learning classifiers that identify the **B**eginning, **I**nside and **L**ast tokens of multi-token chunks as well as **U**nit-length chunks. The problem is transformed into a simple, but constrained, 5-class classification problem.

- Named Entity Recognition Component

  We use existing tools to extract named entities as additional mention head candidates. We choose the state-of-the-art "Illinois Named Entity Tagger" package. This package gives the standard Person/Location/Organization/Misc labels and we take all output named entities as candidates.

- Wikipedia Component

  Many mention heads can be directly matched to a Wikipedia title. We get 4,045,764 Wikipedia titles from Wikipedia dumps and use all of them as potential mention heads. We first run this matching component on training documents and compute the precision of entries that appear in the text (the probability of appearing as mention heads). We then get the set of entries with precision higher than a threshold α, which is tuned on the development set using F1-score. We use them as candidates for mention head matching.

- Known Head Component

  Some mention heads appear repeatedly in the text. To fully utilize the training data, we construct a known mention head candidate set and identify them in the test documents. To balance between recall and precision, we set a parameter $\beta > 0$ as a precision threshold and only allow those mention heads with precision larger than β on the training set. Note that threshold β is also tuned on the development set using F1-score.

We also employ a simple word variation tolerance algorithm in our matching components, to generalize over small variations (plural/singular, etc.).

4.1.4 Mention Head Detection

The mention head detection model is a binary classifier $g_m = w_1^\top \varphi(m)$, in which $\varphi(m)$ is a feature vector for mention head candidate m and w_1 is the corresponding weight vector. We identify a candidate m as a mention head if $g_m > 0$. The features utilized in the vector $\varphi(m)$ consist of:

- Gazetteer features
- Part-Of-Speech features
- Wordnet features
- Features from the previous and next tokens
- Length of mention head
- Normalized Point-wise Mutual Information (NPMI) on the tokens across a mention head boundary
- Feature conjunctions

Altogether there are hundreds of thousands of sparse features.

4.1.5 ILP-based Mention-Pair Coreference

Let M be the set of all mentions. We train a coreference model by learning a pairwise mention scoring function. Specifically, given a mention-pair (u, v) ($u, v \in M$, u is the antecedent of v), we learn a left-linking scoring function $f_{u,v} = w_2^\top \phi(u, v)$, where $\phi(u, v)$ is a pairwise feature vector and w_2 is the weight vector. The inference algorithm is inspired by the best-left-link approach (Chang et al., 2011), where they solve the following ILP problem:

$$\begin{aligned} &\arg\max_{y} \sum_{u<v, u,v \in M} f_{u,v} y_{u,v}, \\ &\text{s.t.} \sum_{u<v} y_{u,v} \leq 1, \quad \forall v \in M, \\ &\qquad y_{u,v} \in \{0, 1\} \quad \forall u, v \in M. \end{aligned} \tag{4.1}$$

Here, $y_{u,v} = 1$ iff mentions u, v are directly linked. Thus, we can construct a forest and the mentions in the same connected component (i.e., in the same tree) are co-referred. For this mention-pair coreference model $\phi(u, v)$, we use the same set of features used in Bengtson and Roth (2008).

4.1.6 Joint Inference Framework

We extend expression (4.1) to facilitate joint inference on mention heads and coreference as follows:

$$\begin{aligned}
&\arg\max_{y} \sum_{u<v, u,v \in M} f_{u,v} y_{u,v} + \sum_{m \in M} g_m y_m, \\
&\text{s.t.} \sum_{u<v} y_{u,v} \leq 1, \quad \forall v \in M', \\
&\quad \sum_{u<v} y_{u,v} \leq y_v, \quad \forall v \in M', \\
&\quad y_{u,v} \in \{0,1\}, \quad y_m \in \{0,1\} \quad \forall u, v, m \in M'.
\end{aligned}$$

Here, M' is the set of all mention head candidates. y_m is the decision variable for mention head candidate m. $y_m = 1$ if and only if the mention head m is chosen. To consider coreference decisions and mention head decisions together, we add the constraint $\sum_{u<v} y_{u,v} \leq y_v$, which ensures that if a candidate mention head v is not chosen, then it will not have coreference links with other mention heads.

4.1.7 Joint Learning Framework

To support joint learning of the parameters w_1 and w_2 described above, we define a joint training objective function $C(w_1, w_2)$ for mention head detection and coreference, which uses a max-margin approach to learn both weight vectors. Suppose we have a collection of documents D, and we generate n_d mention head candidates for each document d ($d \in D$). We use an indicator function $\delta(u, m)$ to represent whether mention heads u, m are in the same coreference cluster based on gold annotations ($\delta(u, m) = 1$ iff they are in the same cluster). Similarly, $\Omega(m)$ is an indicator function representing whether mention head m is valid in the gold annotations.

For simplicity, we first define

$$\begin{aligned}
u' &= \arg\max_{u<m} (w_2^\top \phi(u, m) - \delta(u, m)), \\
u'' &= \arg \max_{u<m, \delta(u,m)=1} w_2^\top \phi(u, m) \Omega(m).
\end{aligned}$$

We then minimize the following joint training objective function $C(w_1, w_2)$.

$$\begin{aligned}
C(w_1, w_2) = \frac{1}{|D|} \sum_{d \in D} \frac{1}{n_d} \sum_{m} (C_{coref,m}(w_2) \\
+ C_{local,m}(w_1) + C_{trans,m}(w_1)) + R(w_1, w_2).
\end{aligned}$$

$C(w_1, w_2)$ is composed of four parts. The first part is the loss function for coreference, where we have

$$C_{coref,m}(w_2) = -w_2^\top \phi(u'', m)\Omega(m) + (w_2^\top \phi(u', m) - \delta(u', m))(\Omega(m) \vee \Omega(u')).$$

It is similar to the loss function for a latent left-linking coreference model[4]. As the second component, we have the quadratic loss for the mention head detection model,

$$C_{local,m}(w_1) = \frac{1}{2}(w_1^\top \varphi(m) - \Omega(m))^2.$$

Using the third component, we further maximize the margin between valid and invalid mention head candidates when they are selected as the best-left-link mention heads for any valid mention head. It can be represented as

$$C_{trans,m}(w_1) = \frac{1}{2}(w_1^\top \varphi(u') - \Omega(u'))^2\Omega(m).$$

The last part is the regularization term

$$R(w_1, w_2) = \frac{\lambda_1}{2}||w_1||^2 + \frac{\lambda_2}{2}||w_2||^2.$$

4.1.8 Stochastic Subgradient Descent for Joint Learning

For joint learning, we choose stochastic subgradient descent (SGD) approach to facilitate performing SGD on a per mention head basis. Next, we describe the weight update algorithm by defining the subgradients.

The partial subgradient w.r.t. mention head m for the head weight vector w_1 is given by

$$\nabla_{w_1,m} C(w_1, w_2) = \frac{1}{|D|n_d}(\nabla C_{local,m}(w_1) + \nabla C_{trans,m}(w_1)) + \lambda_1 w_1, \tag{4.2}$$

where

$$\nabla C_{local,m}(w_1) = (w_1^\top \varphi(m) - \Omega(m))\varphi(m),$$
$$\nabla C_{trans,m}(w_1) = (w_1^\top \varphi(u') - \Omega(u'))\varphi(u')\Omega(m).$$

The partial subgradient w.r.t. mention head m for the coreference weight vector w_2 is given by

$$\nabla_{w_2,m} C(w_1, w_2) = \lambda_2 w_2 + \begin{cases} \phi(u', m) - \phi(u'', m) & \text{if } \Omega(m) = 1, \\ \phi(u', m) & \text{if } \Omega(m) = 0 \text{ and } \Omega(u') = 1, \\ 0 & \text{if } \Omega(m) = 0 \text{ and } \Omega(u') = 0. \end{cases} \tag{4.3}$$

[4]More details can be found in Chang et al. (2013). The difference here is that we also consider the validity of mention heads using $\Omega(u), \Omega(m)$

Here λ_1 and λ_2 are regularization coefficients which are tuned on the development set. To learn the mention head detection model, we consider two different parts of the gradient in expression (4.2). $\nabla C_{local,m}(w_1)$ is exactly the local gradient of mention head m while we add $\nabla C_{trans,m}(w_1)$ to represent the gradient for mention head u', the mention head chosen by the current best-left-linking model for m. This serves to maximize the margin between valid mention heads and invalid ones. As invalid mention heads will not be linked to any other mention head, ∇_{trans} is zero when m is invalid. When training the mention-pair coreference model, we only consider gradients when at least one of the two mention heads m, u' is valid, as shown in expression (4.3). When mention head m is valid ($\Omega(m) = 1$), the gradient is the same as local training for best-left-link of m (first condition in expression (4.3)). When m is not valid while u' is valid, we only demote the coreference link between them (second condition in expression (4.3)). We consider only the gradient from the regularization term when both m, u' are invalid.

As mentioned before, our framework can handle annotations with or without singleton mentions. When the gold data contains no singleton mentions, we have $\Omega(m) = 0$ for all singleton mention heads among mention head candidates. Then, our mention head detection model partly serves as a singleton head detector, and tries to reject singletons in the joint decisions with coreference. When the gold data contains singleton mentions, we have $\Omega(m) = 1$ for all valid singleton mention heads. Our mention head detection model then only learns to differentiate invalid mention heads from valid ones, and thus has the ability to preserve valid singleton heads.

Most of the head mentions are positive examples. We ensure a balanced training of the mention head detection model by adding sub-sampled invalid mention head candidates as negative examples. Specifically, after mention head candidate generation, we train on a set of candidates with precision larger than 50%. We then use Illinois Chunker (Punyakanok and Roth, 2001)[5] to extract more noun phrases from the text and employ Collins head rules (Collins, 1999) to identify their heads. When these extracted heads do not overlap with gold mention heads, we treat them as negative examples.

We note that the aforementioned joint framework can take as input either complete mention candidates or mention head candidates. However, in this paper we only feed mention heads into it. Our experimental results support our intuition that this provides better results.

4.1.9 Experiments

Results on ACE-2004 and CoNLL-2012 datasets show that our system reduces the performance gap for coreference by around 25% (measured as the ratio of performance improvement over performance gap) and

[5] `http://cogcomp.cs.illinois.edu/page/software_view/Chunker`

Systems	MUC	B^3	$CEAF_e$	AVG	MUC	B^3	$CEAF_e$	AVG
	ACE-2004				CoNLL-2012			
$Gold_{M/H}$	78.17	81.64	78.45	79.42	82.03	70.59	66.76	73.12
$Stanford_M$	63.89	70.33	70.21	68.14	64.62	51.89	48.23	54.91
$HotCoref_M$	——	——	——	——	70.74	58.37	55.47	61.53
$Berkeley_M$	——	——	——	——	71.24	58.71	55.18	61.71
$Predicted_M$	64.28	70.37	70.16	68.27	69.63	57.46	53.16	60.08
$H\text{-}M\text{-}Coref_M$	65.81	71.97	71.14	69.64	70.95	59.11	54.98	61.68
$H\text{-}Joint\text{-}M_M$	**67.28**	**73.06**	**73.25**	**71.20**	**72.22**	**60.50**	**56.37**	**63.03**
Clark and Manning (2015)	——	——	——	——	72.6	60.4	56.0	63.0
Wiseman et al. (2015)	——	——	——	——	72.6	60.5	57.1	63.4
Wiseman et al. (2016)	——	——	——	——	73.4	61.5	57.7	64.2
Clark and Manning (2016a)	——	——	——	——	74.6	63.4	59.2	65.7
Lee et al. (2017)	——	——	——	——	77.2	66.6	62.6	68.8
Peters et al. (2018)	——	——	——	——	——	——	——	**70.4**
$Stanford_H$	70.28	73.93	73.04	72.42	68.53	56.68	52.36	59.19
$HotCoref_H$	——	——	——	——	72.94	60.27	57.53	63.58
$Berkeley_H$	——	——	——	——	73.05	60.39	57.43	63.62
$Predicted_H$	71.35	75.33	74.02	73.57	72.11	60.12	55.68	62.64
$H\text{-}M\text{-}Coref_H$	71.81	75.69	74.45	73.98	73.22	61.42	56.21	63.62
$H\text{-}Joint\text{-}M_H$	**72.74**	**76.69**	**75.18**	**74.87**	**74.83**	**62.77**	**57.93**	**65.18**

Table 4.2: Performance of coreference resolution on the ACE-2004 and CoNLL-2012 dataset. Subscripts ($_M$, $_H$) indicate evaluations on (mentions, mention heads) respectively. For gold mentions and mention heads, they yield the same performance for coreference. Our proposed *H-Joint-M* system achieves the highest performance as in June 2015. Parameters of our proposed system are tuned as $\alpha = 0.9$, $\beta = 0.9$, $\lambda_1 = 0.25$ and $\lambda_2 = 0.2$. We also include more recent results where the current state-of-the-art performance is achieved by Peters et al. (2018) using ELMo embeddings.

improves the overall mention detection by over 10 F1 points.

Experimental Setup

We choose three publicly available state-of-the-art end-to-end coreference systems as our baselines: *Stanford* system (Lee et al., 2011), *Berkeley* system (Durrett and Klein, 2014) and *HOTCoref* system (Björkelund and Kuhn, 2014). Our developed system is built on the work by Chang et al. (2013), using Constrained Latent Left-Linking Model (CL^3M) as our mention-pair coreference model in the joint framework[6]. When the CL^3M coreference system uses gold mentions or heads, we call the system *Gold*; when it uses predicted mentions or heads, we call the system *Predicted*. The mention head candidate generation module along with mention boundary detection module can be grouped together to form a complete mention detection system, and we call it *H-M-MD*. We can feed the predicted mentions from *H-M-MD* directly into the mention-pair coreference model that we implemented, resulting in a traditional pipelined end-to-end coreference system, namely *H-M-Coref*. We name the proposed end-to-end coreference system incorporating both the mention head candidate generation module and the joint framework as *H-Joint-M*.

[6]We use Gurobi v5.0.1 as our ILP solver.

Performance for Coreference Resolution

Performance of coreference resolution for all systems on the ACE-2004 and CoNLL-2012 datasets is shown in Table 4.2.[7] These results show that our developed system *H-Joint-M* shows significant improvement on all metrics for both datasets. Existing systems only report results on mentions. Here, we also show their performance evaluated on mention heads. When evaluated on mention heads rather than mentions[8], we can always expect a performance increase for all systems on both datasets. Even though evaluating on mentions is more common in the literature, it is often enough to identify just mention heads in coreference chains. *H-M-Coref* can already bring substantial performance improvement, which indicates that it is helpful for coreference to just identify high quality mention heads. Our proposed *H-Joint-M* system outperforms all baselines and achieves the best results reported at the time of its publication (June 2015). We also include more recent results in Table 4.2. Later systems employ deep learning methods and only report performance results on the CoNLL-2012 data. The current state-of-the-art performance is achieved by Peters et al. (2018) using ELMo embeddings.

Performance for Mention Detection

The performance of mention detection for all systems on the ACE-2004 and CoNLL-2012 datasets is shown in Table 4.3. These results show that our developed system exhibits significant improvement on precision and recall for both datasets. *H-M-MD* mainly improves on recall, indicating, as expected, that the mention head candidate generation module ensures high recall on mention heads. *H-Joint-M* mainly improves on precision, indicating, as expected, that the joint framework correctly rejects many of the invalid mention head candidates during joint inference. Our joint model can adapt to annotations with or without singleton mentions. Based on training data, our system has the ability to preserve true singleton mentions in ACE while rejecting many singleton mentions in OntoNotes.[9]. Note that we have better mention detection results on ACE-2004 dataset than on OntoNotes-5.0 dataset. We believe that this is due to the fact that extracting mention heads in the OntoNotes dataset is somewhat noisy.

Analysis of Performance Improvement

The improvement of our *H-Joint-M* system is due to two distinct but related modules: the mention head candidate generation module ("Head") and the joint learning and inference framework ("Joint").[10] We evaluate the effect of these two modules in terms of *Mention Detection Error Reduction* (MDER) and *Performance*

[7]We do not provide results from *Berkeley* and *HOTCoref* on ACE-2004 dataset as they do not directly support ACE input. Results for *HOTCoref* are slightly different from the results reported in Björkelund and Kuhn (2014). For *Berkeley* system, we use the reported results from Durrett and Klein (2014).

[8]Here, we treat mention heads as mentions. Thus, in the evaluation script, we set the boundary of a mention to be the boundary of its corresponding mention head.

[9]Please note that when evaluating on OntoNotes, we eventually remove all singleton mentions from the output.

[10]"Joint" rows are computed as "H-Joint-M" rows minus "Head" rows. They reflect the contribution of the joint framework to mention detection (by rejecting some mention heads).

Systems	Precision	Recall	F1-score	Precision	Recall	F1-score
	ACE-2004			CoNLL-2012		
Predicted$_M$	75.11	73.03	74.06	65.28	63.41	64.33
H-M-MD$_M$	77.45	92.97	83.90	70.09	76.72	73.26
H-Joint-M$_M$	85.34	91.73	**88.42**	78.51	75.52	**76.99**
Predicted$_H$	76.84	86.99	79.87	76.38	74.02	75.18
H-M-MD$_H$	80.82	93.45	86.68	77.73	83.99	80.74
H-Joint-M$_H$	88.85	92.27	**90.53**	85.07	82.31	**83.67**

Table 4.3: Performance of mention detection on the ACE-2004 and CoNLL-2012 datasets. Subscripts ($_M$, $_H$) indicate evaluations on (mentions, mention heads) respectively.

ACE-2004	MDER	PGR(AVG)
Head$_M$	37.93	12.29
Joint$_M$	17.43	13.99
H-Joint-M$_M$	55.36	26.28
Head$_H$	34.00	7.01
Joint$_H$	19.22	15.21
H-Joint-M$_H$	53.22	22.22

CoNLL-2012	MDER	PGR(AVG)
Head$_M$	25.04	12.16
Joint$_M$	10.45	10.44
H-Joint-M$_M$	35.49	22.60
Head$_H$	22.40	10.58
Joint$_H$	11.81	13.75
H-Joint-M$_H$	34.21	24.33

Table 4.4: Analysis of performance improvement in terms of *Mention Detection Error Reduction* (MDER) and *Performance Gap Reduction* (PGR) for coreference resolution on the ACE-2004 and CoNLL-2012 datasets. "Head" represents the mention head candidate generation module, "Joint" represents the joint learning and inference framework, and "H-Joint-M" indicates the end-to-end system.

Gap Reduction (PGR) for coreference. MDER is computed as the ratio of performance improvement for mention detection over the original mention detection error rate, while PGR is computed as the ratio of performance improvement for coreference over the performance gap for coreference. Results on the ACE-2004 and CoNLL-2012 datasets are shown in Table 4.4.[11]

The mention head candidate generation module has a bigger impact on MDER compared to the joint framework. However, they both have the same level of positive effects on PGR for coreference resolution. On both datasets, we achieve more than 20% performance gap reduction for coreference.

[11] We use bootstrapping resampling (10 times from the test data) with signed rank test. All the improvements shown are statistically significant.

4.2 Solving Hard Co-reference Problems

4.2.1 Motivation

One fundamental difficulty has been that of resolving instances involving pronouns since they often require deep language understanding and use of background knowledge. Existing methods perform particularly poorly on pronouns, specifically when gender or plurality information cannot help. In this paper, we aim to improve coreference resolution by addressing these hard problems. Consider the following examples:

Ex.1 *[A bird]*$_{e_1}$ *perched on the [limb]*$_{e_2}$ *and [it]*$_{pro}$ *bent.*
Ex.2 *[Robert]*$_{e_1}$ *was robbed by [Kevin]*$_{e_2}$*, and [he]*$_{pro}$ *is arrested by police.*

In both examples, one cannot resolve the pronouns based on only gender or plurality information. Recently, Rahman and Ng (2012) gathered a dataset containing 1886 sentences of such challenging pronoun resolution problems (referred to later as the *Winograd* dataset, following Winograd (1972) and Levesque et al. (2011)). As an indication to the difficulty of these instances, we note that a state-of-the-art coreference resolution system (Chang et al., 2013) achieves precision of 53.26% on it. A special purpose classifier (Rahman and Ng, 2012) trained on this data set achieves 73.05%. The key contribution of this paper is a general purpose, state-of-the-art coreference approach which, at the same time, achieves precision of 76.76% on these hard cases.

4.2.2 System Design

Addressing these hard coreference problems requires significant amounts of background knowledge, along with an inference paradigm that can make use of it in supporting the coreference decision. Specifically, in Ex.1 one needs to know that "a limb bends" is more likely than "a bird bends". In Ex.2 one needs to know that the *subject* of the verb "rob" is more likely to be the *object* of "arrest" than the *object* of the verb "rob" is. The knowledge required is, naturally, centered around the key predicates in the sentence, motivating the central notion proposed in this paper, that of *Predicate Schemas*. In this chapter, we develop the notion of *Predicate Schemas*, instantiate them with automatically acquired knowledge, and show how to compile it into constraints that are used to resolve coreference within a general *Integer Linear Programming* (ILP) driven approach to coreference resolution. Specifically, we study two types of Predicate Schemas that, as we show, cover a large fraction of the challenging cases. The first specifies one predicate with its subject and object, thus providing information on the subject and object preferences of a given predicate. The second specifies two predicates with a semantically shared argument (either subject or object), thus specifies role

preferences of one predicate, among roles of the other. We instantiate these schemas by acquiring statistics in an unsupervised way from multiple resources including the Gigaword corpus, Wikipedia, Web Queries and polarity information.

In this work, we propose an algorithmic solution that involves a new representation for the knowledge required to address hard coreference problems, along with a constrained optimization framework that uses this knowledge in coreference decision making. Our representation, Predicate Schemas, is instantiated with knowledge acquired in an unsupervised way, and is compiled automatically into constraints that impact the coreference decision. We present a general coreference resolution system that significantly improves state-of-the-art performance on hard, *Winograd*-style, pronoun resolution cases, while still performing at the state-of-the-art level on standard coreference resolution datasets. More details can be referred to the published work in Peng et al. (2015b).

Coreference resolution is one of the most important tasks in *Natural Language Processing* (NLP). Although there is a plethora of works on this task (Soon et al., 2001; Ng and Cardie, 2002a; Ng, 2004; Bengtson and Roth, 2008; Pradhan et al., 2012; Kummerfeld and Klein, 2013; Chang et al., 2013), it is still deemed an unsolved problem due to intricate and ambiguous nature of natural language text.

A lot of recent work has attempted to utilize similar types of resources to improve coreference resolution (Rahman and Ng, 2011; Ratinov and Roth, 2012; Bansal and Klein, 2012; Rahman and Ng, 2012). The common approach has been to inject knowledge as features. However, these pieces of knowledge provide relatively strong evidence that loses impact in standard training due to sparsity. Instead, we compile our Predicate Schemas knowledge automatically, at inference time, into constraints, and make use of an ILP driven framework (Roth and Yih, 2004) to make decisions. Using constraints is also beneficial when the interaction between multiple pronouns is taken into account when making global decisions. Consider the following example:

Ex.3 *[Jack]*$_{e_1}$ *threw the bags of [John]*$_{e_2}$ *into the water since [he]*$_{pro_1}$ *mistakenly asked [him]*$_{pro_2}$ *to carry [his]*$_{pro_3}$ *bags.*

In order to correctly resolve the pronouns in Ex.3, one needs to have the knowledge that "*he* <u>asks</u> *him*" indicates that *he* and *him* refer to different entities (because they are subject and object of the same predicate; otherwise, *himself* should be used instead of *him*). This knowledge, which can be easily represented as constraints during inference, then impacts other pronoun decisions in a global decision with respect to all pronouns: pro_3 is likely to be different from pro_2, and is likely to refer to e_2. This type of inference can be easily represented as a constraint during inference, but hard to inject as a feature.

Category	#	Sentence
1	1.1	*[The bird]$_{e_1}$ perched on the [limb]$_{e_2}$ and [it]$_{pro}$ bent.*
	1.2	*[The bee]$_{e_1}$ landed on [the flower]$_{e_2}$ because [it]$_{pro}$ had pollen.*
2	2.1	*[Bill]$_{e_1}$ was robbed by [John]$_{e_2}$, so the officer arrested [him]$_{pro}$.*
	2.2	*[Jimbo]$_{e_1}$ was afraid of [Bobbert]$_{e_2}$ because [he]$_{pro}$ gets scared around new people.*
3	3.1	*[Lakshman]$_{e_1}$ asked [Vivan]$_{e_2}$ to get him some ice cream because [he]$_{pro}$ was hot.*
	3.2	*Paula liked [Ness]$_{e_1}$ more than [Pokey]$_{e_2}$ because [he]$_{pro}$ was mean to her.*

Table 4.5: Example sentences for each schema category. The annotated entities and pronouns are hard coreference problems.

Type	Schema form	Explanation of examples from Table 4.5
1	$pred_m(m, a)$	Example 1.2: It is enough to know that: $\mathcal{S}(\text{have}(m = [\text{the flower}], a = [\text{pollen}])) > \mathcal{S}(\text{have}(m = [\text{the bee}], a = [\text{pollen}]))$
2	$pred_m(m, a) \mid \widehat{pred}_m(m, \widehat{a}), cn$	Example 2.2: It is enough to know that: $\mathcal{S}(\textit{be afraid of}(m = *, a = *) \mid \textit{get scared}(m = *, \widehat{a} = *), \text{because}) > \mathcal{S}(\textit{be afraid of}(a = *, m = *) \mid \textit{get scared}(m = *, \widehat{a} = *), \text{because})$

Table 4.6: Predicate Schemas and examples of the logic behind the schema design. Here $*$ indicates that the argument is dropped, and $\mathcal{S}(.)$ denotes the scoring function defined in the text.

The main contributions can be summarized as follows:

1. We propose the Predicate Schemas representation and study two specific schemas that are important for coreference.

2. We show how, in a given context, Predicate Schemas can be automatically compiled into constraints and affect inference.

3. Consequently, we address hard pronoun resolution problems as a standard coreference problem and develop a system which shows significant improvement for hard coreference problems while achieving the same state-of-the-art level of performance on standard coreference problems.

4.2.3 Predicate Schema

We present multiple kinds of knowledge that are needed in order to improve hard coreference problems. Table 4.5 provides two example sentences for each type of knowledge. We use m to refer to a mention. A mention can either be an entity e or a pronoun pro. $pred_m$ denotes the predicate of m (similarly, $pred_{pro}$ and $pred_e$ for pronouns and entities, respectively). For instance, in sentence 1.1 in Table 4.5, the predicate of e_1 and e_2 is $pred_{e_1} = pred_{e_2} =$"perch on". cn refers to the discourse connective (cn="and" in sentence 1.1). a denotes an argument of $pred_m$ other than m. For example, in sentence 1.1, assuming that $m = e_1$, the corresponding argument is $a = e_2$.

Type 1	$\mathcal{S}\left(pred_m\left(m,a\right)\right)$ $\mathcal{S}\left(pred_m\left(a,m\right)\right)$ $\mathcal{S}\left(pred_m\left(m,*\right)\right)$ $\mathcal{S}\left(pred_m\left(*,m\right)\right)$
Type 2	$\mathcal{S}\left(pred_m\left(m,a\right)\left\|\widehat{pred}_m\left(m,\widehat{a}\right),cn\right.\right)$ $\mathcal{S}\left(pred_m\left(a,m\right)\left\|\widehat{pred}_m\left(m,\widehat{a}\right),cn\right.\right)$ $\mathcal{S}\left(pred_m\left(m,a\right)\left\|\widehat{pred}_m\left(\widehat{a},m\right),cn\right.\right)$ $\mathcal{S}\left(pred_m\left(a,m\right)\left\|\widehat{pred}_m\left(\widehat{a},m\right),cn\right.\right)$ $\mathcal{S}\left(pred_m\left(m,*\right)\left\|\widehat{pred}_m\left(m,*\right),cn\right.\right)$ $\vdots$

Table 4.7: Possible variations for scoring function statistics. Here $*$ indicates that the argument is dropped.

We represent the knowledge needed with two types of Predicate Schemas (as depicted in Table 4.6). To solve the assignment of $[\text{it}]_{pro}$ in sentence 1.1, we need the knowledge that "a limb bends" is more reasonable than "a bird bends". Note that the predicate of the pronoun is playing a key role here. Also the entity mention itself is essential. Similarly, for sentence 1.2, to resolve $[\text{it}]_{pro}$, we need the knowledge that "flower had pollen" is more reasonable than "bee had pollen". Here, in addition to entity mention and the predicate (of the pronoun), we need the argument which shares the predicate with the pronoun. To formally define the type of knowledge needed we denote it with "$\boldsymbol{pred_m(m,a)}$" where m and a are a mention and an argument, respectively[12]. We use $\mathcal{S}(.)$ to denote the score representing how likely the combination of the predicate-mention-argument is. For each schema, we use several variations by either changing the order of the arguments (*subj.* vs *obj.*) or dropping either of them. We score the various Type 1 and Type 2 schemas (shown in Table 4.7) differently. The first row of Table 4.6 shows how Type 1 schema is being used in the case of Sentence 1.2.

For sentence 2.2, we need to have the knowledge that the *subject* of the verb phrase "be afraid of" is more likely than the *object* of the verb phrase "be afraid of" to be the *subject* of the verb phrase "get scared". The structure here is more complicated than that of Type 1 schema. To make it clearer, we analyze sentence 2.1. In this sentence, the *object* of "be robbed by" is more likely than the *subject* of the verb phrase "be robbed by" to be the *object* of "the officer arrest". We can see in both examples (and for the Type 2 schema in general), that both predicates (the entity predicate and the pronoun predicate) play a crucial role. Consequently, we design the Type 2 schema to capture the interaction between the entity predicate and the pronoun predicate. In addition to the predicates, we may need mention-argument information. Also, we

[12]Note that the order of m and a relative to the predicate is a critical issue. To keep things general in the schemas definition, we do not show the ordering; however, when using scores in practice the order between a mention and an argument is a critical issue.

stress the importance of the discourse connective between entity mention and pronoun; if in either sentence 2.1 or 2.2, we change the discourse connective to "although", the coreference resolution will completely change. Overall, we can represent the knowledge as "$\boldsymbol{pred_m\,(m,a)\,|\widehat{pred}_m\,(m,\widehat{a})\,,cn}$". Just like for Type 1 schema, we can represent Type 2 schema with a score function for different variations of arguments (lower half of Table 4.7). In Table 4.6, we exhibit this for sentence 2.2.

Type 3 contains the set of instances which cannot be solved using schemas of Type 1 or 2. Two such examples are included in Table 4.5. In sentence 3.1 and 3.2, the context containing the necessary information goes beyond our triple representation and therefore this instance cannot be resolved with either of the two schema types. It is important to note that the notion of Predicate Schemas is more general than the Type 1 and Type 2 schemas introduced here. Designing more informative and structured schemas will be essential to resolving additional types of hard coreference instances.

4.2.4 Constrained ILP Inference

Integer Linear Programming (ILP) based formulations of NLP problems (Roth and Yih, 2004) have been used in a board range of NLP problems and, particularly, in coreference problems (Chang et al., 2011; Denis and Baldridge, 2007). Our formulation is inspired by Chang et al. (2013). Let $\mathcal{M}$ be the set of all mentions in a given text snippet, and $\mathcal{P}$ the set of all pronouns, such that $\mathcal{P} \subset \mathcal{M}$. We train a coreference model by learning a pairwise mention scoring function. Specifically, given a mention-pair $(u, v) \in \mathcal{M}$ (u is the antecedent of v), we learn a left-linking scoring function $f_{u,v} = \mathbf{w}^\top \phi(u, v)$, where $\phi(u, v)$ is a pairwise feature vector and $\mathbf{w}$ is the weight vector. We follow the *Best-Link* approach (Section 2.3 from Chang et al. (2011)). The ILP problem that we solve is formally defined as follows:

$$\begin{cases} \arg\max_{y} \sum_{u \in \mathcal{M}, v \in \mathcal{M}} f_{u,v} y_{u,v} \\ \text{s.t. } \; y_{u,v} \in \{0, 1\}, \quad \forall u, v \in \mathcal{M} \\ \qquad \sum_{u<v, u \in \mathcal{M}} y_{u,v} \leq 1, \quad \forall v \in \mathcal{M} \\ \qquad \text{Constraints from Predicate Schemas Knowledge} \\ \qquad \text{Constraints between pronouns.} \end{cases} \tag{4.4}$$

Here, u, v are mentions and $y_{u,v}$ is the decision variable to indicate whether or not mention u and mention v are coreferents. As the first constraint shows, $y_{u,v}$ is a binary variable. $y_{u,v}$ equals 1 if u, v are coreferents and 0 otherwise. The second constraint indicates that we only choose at most one antecedent to be coreferent

with each mention v. ($u < v$ represents that u appears before v, thus u is an antecedent of v.) In this work, we add constraints from Predicate Schemas Knowledge and between pronouns.

The Predicate Schemas knowledge provides a vector of score values $\mathcal{S}(u, v)$ for mention pairs $\{(u, v)|(u \in \mathcal{M}, v \in \mathcal{P}\}$, which concatenates all the schemas involving u and v. Entries in the score vector are designed so that the larger the value is, the more likely u and v are to be coreferents. We have two ways to use the score values: 1) Augmenting the feature vector $\phi(u, v)$ with these scores. 2) Casting the scores as constraints for the coreference resolution ILP in one of the following forms:

$$\begin{cases} \text{if } s_i(u, v) \geq \alpha_i s_i(w, v) \Rightarrow y_{u,v} \geq y_{w,v}, \\ \text{if } s_i(u, v) \geq s_i(w, v) + \beta_i \Rightarrow y_{u,v} \geq y_{w,v}, \end{cases} \tag{4.5}$$

where $s_i(.)$ is the i-th dimension of the score vector $\mathcal{S}(.)$ corresponding to the i-th schema represented for a given mention pair. α_i and β_i are threshold values which we tune on a development set.[13] If an inequality holds for all relevant schemas (that is, all the dimensions of the score vector), we add an inequality between the corresponding indicator variables inside the ILP. [14] As we increase the value of a threshold, the constraints in (4.5) become more conservative, thus it leads to fewer but more reliable constraints added into the ILP. We tune the threshold values such that their corresponding scores attain high enough accuracy, either in the multiplicative form or the additive form.[15] Note that, given a pair of mentions and context, we automatically instantiate a collection of relevant schemas, and then generate and evaluate a set of corresponding constraints. To the best of our knowledge, this is the first work to use such automatic constraint generation and tuning method for coreference resolution with ILP inference. Next we describe how we acquire the score vectors $\mathcal{S}(u, v)$ for the Predicate Schemas in an unsupervised fashion.

We now briefly explain the pre-processing step required in order to extract the score vector $\mathcal{S}(u, v)$ from a pair of mentions. Define a triple structure $t_m \triangleq pred_m(m, a_m)$ for any $m \in \mathcal{M}$. The subscript m for *pred* and a, emphasizes that they are extracted as a function of the mention m. The extraction of triples is done by utilizing the dependency parse tree from the Easy-first dependency parser (Goldberg and Elhadad, 2010). We start with a mention m, and extract its related predicate and the other argument based on the dependency parse tree and part-of-speech information. To handle multi-word predicates and arguments, we use a set of hand-designed rules. We then get the score vector $\mathcal{S}(u, v)$ by concatenating all scores of the Predicate Schemas given two triples t_u, t_v. Thus, we can expand the score representation for each type of Predicate Schemas given in Table 4.6: 1) For Type 1 schema, $\mathcal{S}(u, v) \equiv \mathcal{S}(pred_v(m = u, a = a_v))$ [16] 2) For Type 2 schema, $\mathcal{S}(u, v) \equiv \mathcal{S}(pred_u(m = u, a = a_u)|\widehat{pred}_v(m = v, a = a_v), cn)$.

[13] For the i_{th} dimension of the score vector, we choose either α_i or β_i as the threshold.
[14] If the constraints dictated by any two dimensions of $\mathcal{S}$ are contradictory, we ignore both of them.
[15] The choice is made based on the performance on the development set.
[16] In $pred_v(m = u, a = a_v)$ the argument and the predicate are extracted relative to v but the mention m is set to be u.

$$s_{pol}(u,v) = \begin{bmatrix} \mathbf{1}\{Po(p_u) = + \text{ AND } Po(p_v) = +\} \text{ OR } \mathbf{1}\{Po(p_u) = - \text{ AND } Po(p_v) = -\} \\ \mathbf{1}\{Po(p_u) = + \text{ AND } Po(p_v) = +\} \\ \mathbf{1}\{Po(p_u) = - \text{ AND } Po(p_v) = -\} \end{bmatrix}$$

Table 4.8: Extracting the polarity score given polarity information of a mention-pair (u, v). To be brief, we use the shorthand notation $p_v \triangleq pred_v$ and $p_u \triangleq pred_u$. $\mathbf{1}\{\cdot\}$ is an indicator function. $s_{pol}(u, v)$ is a binary vector of size three.

In additional to schema-driven constraints, we also apply constraints between pairs of pronouns within a fixed distance[17]. For two pronouns that are semantically different (e.g. *he* vs. *it*), they must refer to different antecedents. For two non-possessive pronouns that are related to the same predicate (e.g. *he saw him*), they must refer to different antecedents.[18]

We then incorporate all constraints into a general coreference system (Chang et al., 2013) utilizing the mention-pair model (Ng and Cardie, 2002b; Bengtson and Roth, 2008; Stoyanov et al., 2010). A classifier learns a pairwise metric between mentions, and during inference, we follow the framework proposed in Chang et al. (2011) using ILP.

4.2.5 Knowledge Acquisition

One key point that remains to be explained is how to acquire the knowledge scores $\mathcal{S}(u, v)$. In this section, we propose multiple ways to acquire these scores. We make use of four resources. Each of them generates its own score vector. Therefore, the overall score vector is the concatenation of the score vector from each resource.

$\mathcal{S}(u, v) = [\mathcal{S}_{giga}(u, v)\ \mathcal{S}_{wiki}(u, v)\ \mathcal{S}_{web}(u, v)\ \mathcal{S}_{pol}(u, v)]$.

- Gigaword Co-occurence

  We extract triples $t_m \triangleq pred_m(m, a_m)$ from Gigaword data (4,111,240 documents). We start by extracting noun phrases using the Illinois-Chunker (Punyakanok and Roth, 2001). For each noun phrase, we extract its head noun and then extract the associated predicate and argument to form a triple. We gather the statistics for both schema types after applying lemmatization on the predicates and arguments.

  Using the extracted triples, we get a score vector from each schema type: $\mathcal{S}_{giga} = [\mathcal{S}^{(1)}_{giga}\ \mathcal{S}^{(2)}_{giga}]$. To extract scores for Type 1 Predicate Schemas, we create occurrence counts for each schema instance. After all scores are gathered, our goal is to query $\mathcal{S}^{(1)}_{giga}(u, v) \equiv \mathcal{S}(pred_v(m = u, a = a_v))$ from our knowledge base. The returned score is the $log(.)$ of the number of occurrences. For Type 2 Predicate Schemas, we gather the statistics of triple co-occurrence. We count the co-occurrence of neighboring

[17] We set the distance to be 3 sentences.

[18] Three cases are considered: *he-him*, *she-her*, *they-them*

triples that share at least one linked argument. We consider two triples to be neighbors if they are within a distance of three sentences. We use two heuristic rules to decide whether a pair of arguments between two neighboring triples are coreferents or not: 1) If the head noun of two arguments can match, we consider them coreferents. 2) If one argument in the first triple is a person name and there is a compatible pronoun (based on its gender and plurality information) in the second triple, they are also labeled as coreferents. We also extract the discourse connectives between triples (*because*, *therefore*, etc.) if there are any. To avoid sparsity, we only keep the mention roles (only *subj* or *obj*; no exact strings are kept). Two triple-pairs are considered different if they have different predicates, different roles, different coreferred argument-pairs, or different discourse connectives. The co-occurrence counts extracted in this form correspond to Type 2 schemas in Table 4.6. During inference, we match a Type 2 schema for $\mathcal{S}^{(2)}_{giga}(u,v) \equiv \mathcal{S}(pred_u(m=u, a=a_u)|\widehat{pred}_v(m=u, a=a_v), cn)$. Our method is related, but different from the proposal in Balasubramanian et al. (2012), who suggested to extract triples using an OpenIE system (Mausam et al., 2012). We extracted triples by starting from a mention, then extract the predicate and the other argument. An OpenIE system does not easily provide this ability. Our Gigaword counts are gathered also in a way similar to what has been proposed in Chambers and Jurafsky (2009a), but we gather much larger amounts of data.

- Wikipedia Disambiguated Co-occurence

  We use the publicly available Illinois Wikifier (Cheng and Roth, 2013; Ratinov et al., 2011), a system that disambiguates mentions by mapping them into correct Wikipedia pages, to process the Wikipedia data. We then extract from the Wikipedia text all entities, verbs and nouns, and gather co-occurrence statistics with these syntactic *variations*: 1) *immediately after* 2) *immediately before* 3) *before* 4) *after*. For each of these variations, we get the probability and count[19] of a pair of words (e.g. probability[20]/count for "bend" *immediately following* "limb") as separate dimensions of the score vector. Given the co-occurrence information, we get a score vector $\mathcal{S}_{wiki}(u,v)$ corresponding to Type 1 Predicate Schemas, and hence $\mathcal{S}(u,v)_{wiki} \equiv \mathcal{S}(pred_v(m=u, a=a_v))$.

- Web Search Query Count

  Our third source of score vectors is web queries that we implement using Google queries. We extract a score vector $\mathcal{S}_{web}(u,v) \equiv \mathcal{S}(pred_v(m=u, a=a_v))$ (Type 1 Predicate Schemas) by querying for 1) "u a_v" 2) "u $pred_v$" 3) "u $pred_v$ a_v" 4) "a_v u"[21]. For each variation of nouns (plural and singular) and verbs (different tenses) we create a different query and average the counts over all queries. Concatenating

[19] We use the $log(.)$ of the counts here.
[20] Conditional probability of "limb" immediately following the given verb "bend".
[21] We query this only when a_v is an adjective and $pred_v$ is a to-be verb.

	# Doc	# Train	# Test	# Mention	# Pronoun	# Predictions for Pronoun
Winograd	1886	1212	674	5658	1886	1348
WinoCoref	1886	1212	674	6404	2595	2118
ACE	375	268	107	23247	3862	13836
OntoNotes	3150	2802	348	175324	58952	37846

Table 4.9: Statistics of *Winograd*, *WinoCoref*, *ACE* and *OntoNotes* datasets. We give the total number of mentions and pronouns, while the number of predictions for pronoun is specific for the test data. We added 746 mentions (709 among them are pronouns) to *WinoCoref* compared to *Winograd*.

the counts (each is a separate dimension) would give us the score vector $\mathcal{S}_{web}(u, v)$.

- Polarity of Context

  Another rich source of information is the polarity of context, which has been previously used for Winograd schema problems (Rahman and Ng, 2012). Here we use a slightly modified version. The polarity scores are used for Type 1 Predicate Schemas and therefore we want to get $\mathcal{S}_{pol}(u, v) \equiv \mathcal{S}(pred_v(m = u, a = a_v))$. We first extract polarity values for $Po(pred_u)$ and $Po(pred_v)$ by repeating the following procedures for each of them: 1) We extract initial polarity information given the predicate (using the data provided by Wilson et al. (2005a)). 2) If the role of the mention is *object*, we negate its polarity. 3) If there is a polarity-reversing discourse connective (such as "but") preceding the predicate, we reverse the polarity. 4) If there is a negative comparative adverb (such as "less", "lower") we reverse the polarity. Given the polarity values $Po(pred_u)$ and $Po(pred_v)$, we construct the score vector $\mathcal{S}_{pol}(u, v)$ following Table 4.8.

4.2.6 Experiments

We evaluate our system for both hard coreference problems and general coreference problems, and provide detailed analysis on the impact of our proposed Predicate Schemas. Since we treat resolving hard pronouns as part of the general coreference problems, we extend the *Winograd* dataset with a more complete annotation to get a new dataset. We evaluate our system on both datasets, and show significant improvement over the baseline system and over the results reported in Rahman and Ng (2012). Moreover, we show that, at the same time, our system achieves the state-of-art performance on standard coreference datasets.

Experimental Setup

Since we aim to solve hard coreference problems, we choose to test our system on the *Winograd* dataset[22] (Rahman and Ng, 2012). It is a challenging pronoun resolution dataset which consists of sentence pairs based on Winograd schemas. The original annotation only specifies one pronoun and two entities in each sentence,

[22] Available at http://www.hlt.utdallas.edu/~vince/data/emnlp12/

and it is considered as a binary decision for each pronoun. As our target is to model and solve them as general coreference problems, we expand the annotation to include all pronouns and their linked entities as mentions (We call this new re-annotated dataset *WinoCoref*. Ex.3 is from the *Winograd* dataset. It originally only specifies *he* as the pronoun in question, and we added *him* and *his* as additional target pronouns. We also use two standard coreference resolution datasets *ACE*(2004) (NIST, 2004) and *OntoNotes-5.0* (Pradhan et al., 2011) for evaluation. Statistics of the datasets are provided in Table 4.9. We use the state-of-art Illinois coreference system as our baseline system (Chang et al., 2013). It includes two different versions. One employs *Best-Left-Link* (BLL) inference method (Ng and Cardie, 2002b), and we name it *Illinois*[23]; while the other uses ILP with constraints for inference, and we name it *IlliCons*. Both systems use *Best-Link Mention-Pair* (BLMP) model for training. On *Winograd* dataset, we also treat the reported result from Rahman and Ng (2012) as a baseline. We present three variations of the Predicate Schemas based system developed here. We inject Predicate Schemas knowledge as mention-pair features and retrain the system (*KnowFeat*). We use the original coreference model and Predicate Schemas knowledge as constraints during inference (*KnowCons*). We also have a combined system (*KnowComb*), which uses the schema knowledge to add features for learning as well as constraints for inference. A summary of all systems is provided in Table 4.10.

Evaluation Metrics

When evaluating on the full datasets of *ACE* and *OntoNotes*, we use the widely recognized metrics MUC (Vilain et al., 1995), BCUB (Bagga and Baldwin, 1998), Entity-based CEAF (CEAF_e) (Luo, 2005) and their average. As *Winograd* is a pronoun resolution dataset, we use precision as the evaluation metric. Although *WinoCoref* is more general, each coreferent cluster only contains 2-4 mentions and all are within the same sentence. Since traditional coreference metrics cannot serve as good metrics, we extend the precision metric and design a new one called *AntePre*. Suppose there are k pronouns in the dataset, and each pronoun has $n_1, n_2, \cdots, n_k$ antecedents, respectively. We can view predicted coreference clusters as binary decisions on each antecedent-pronoun pair (linked or not). The total number of binary decisions is $\sum_{i=1}^{k} n_i$. We then measure how many binary decisions among them are correct; let m be the number of correct decisions, then *AnrePre* is computed as: $\frac{m}{\sum_{i=1}^{k} n_i}$.

Results for Hard Coreference Problems

Performance results on *Winograd* and *WinoCoref* datasets are shown in Table 4.11. The best performing system is *KnowComb*. It improves by over 20% over a state-of-art general coreference system on *Winograd* and also outperforms Rahman and Ng (2012) by a margin of 3.3%. On the *WinoCoref* dataset, it improves

[23]In implementation, we use the L^3M model proposed in Chang et al. (2013), which is slightly different. It can be seen as an extension of BLL inference method.

Systems	Learning Method	Inference Method
Illinois	BLMP	BLL
IlliCons	BLMP	ILP
KnowFeat	BLMP+SF	BLL
KnowCons	BLMP	ILP+SC
KnowComb	BLMP+SF	ILP+SC

Table 4.10: Summary of learning and inference methods for all systems. SF stands for schema features while SC represents constraints from schema knowledge.

Dataset	Metric	Illinois	IlliCons	Rahman and Ng (2012)	KnowFeat	KnowCons	KnowComb
Winograd	Precision	51.48	53.26	73.05	71.81	74.93	**76.41**
WinoCoref	AntePre	68.37	74.32	—	88.48	88.95	**89.32**

Table 4.11: Performance results on *Winograd* and *WinoCoref* datasets. All our three systems are trained on *WinoCoref*, and we evaluate the predictions on both datasets. Our systems improve over the baselines by over than 20% on *Winograd* and over 15% on *WinoCoref*.

by 15%. These results show significant performance improvement by using Predicate Schemas knowledge on hard coreference problems. Note that the system developed in Rahman and Ng (2012) cannot be used on the *WinoCoref* dataset. The results also show that it is better to compile knowledge into constraints when the knowledge quality is high than add them as features.

Results for Standard Coreference Problems

Performance results on standard *ACE* and *OntoNotes* datasets are shown in Table 4.12. Our *KnowComb* system achieves the same level of performance as does the state-of-art general coreference system we base it on. As hard coreference problems are rare in standard coreference datasets, we do not have significant performance improvement. However, these results show that our additional Predicate Schemas do not harm the predictions for regular mentions.

Detailed Analysis

To study the coverage of our Predicate Schemas knowledge, we label the instances in *Winograd* (which also applies to *WinoCoref*) with the type of Predicate Schemas knowledge required. The distribution of the instances is shown in Table 4.13. Our proposed Predicate Schemas cover 73% of the instances.

System	MUC	BCUB	CEAFe	AVG
ACE				
IlliCons	**78.17**	81.64	**78.45**	**79.42**
KnowComb	77.51	**81.97**	77.44	78.97
OntoNotes				
IlliCons	84.10	**78.30**	**68.74**	**77.05**
KnowComb	**84.33**	78.02	67.95	76.76

Table 4.12: Performance results on *ACE* and *OntoNotes* datasets. Our system gets the same level of performance compared to a state-of-art general coreference system.

Category	Cat1	Cat2	Cat3
Size	317	1060	509
Portion	16.8%	56.2%	27.0%

Table 4.13: Distribution of instances in *Winograd* dataset of each category. Cat1/Cat2 is the subset of instances that require Type 1/Type 2 schema knowledge, respectively. All other instances are put into Cat3. Cat1 and Cat2 instances can be covered by our proposed Predicate Schemas.

Schema	AntePre(Test)	AntePre(Train)
Type 1	76.67	86.79
Type 2	79.55	88.86
Type 1 (Cat1)	90.26	93.64
Type 2 (Cat2)	83.38	92.49

Table 4.14: Ablation Study of Knowledge Schemas on *WinoCoref*. The first line specifies the performance for *KnowComb* with only Type 1 schema knowledge tested on all data while the third line specifies the performance using the same model but tested on Cat1 data. The second line specifies the performance results for *KnowComb* system with only Type 2 schema knowledge on all data while the fourth line specifies the performance using the same model but tested on Cat2 data.

We also provide an ablation study on the *WinoCoref* dataset in Table 4.14. These results use the best performing *KnowComb* system. They show that both Type 1 and Type 2 schema knowledge have higher precision on Category 1 and Category 2 data instances, respectively, compared to that on full data. Type 1 and Type 2 knowledge have similar performance on full data, but the results show that it is harder to solve instances in category 2 than those in category 1. Also, the performance drop between Cat1/Cat2 and full data indicates that there is a need to design more complicated knowledge schemas and to refine the knowledge acquisition for further performance improvement.

Chapter 5

Event Coreference Resolution

Understanding events necessitates understanding relations among them and, as a minimum, determining whether two snippets of text represent the same event or not – event coreference problem.

5.1 System Design

For event coreference, we follow the MSEP framework discussed in Chapter 3.1 and measure similarities between event mentions in an unsupervised fashion. Similar to the mention-pair model in entity co-reference (Ng and Cardie, 2002b; Bengtson and Roth, 2008; Stoyanov et al., 2010), we use cosine similarities computed from pairs of event mentions: $S(e_1, e_2)$ (as in Eq. (3.1)).

Before applying the co-reference model, we first use external knowledge bases to identify conflict events. We use the Illinois Wikification (Cheng and Roth, 2013) tool to link event arguments to Wikipedia pages. Using the Wikipedia IDs, we map event arguments to Freebase entries. We view the top-level Freebase type as the event argument type. An event argument can contain multiple wikified entities, leading to multiple Wikipedia pages and thus a set of Freebase types. We also augment the argument type set with NER labels: PER (person) and ORG (organization). We add either of the NER labels if we detect such a named entity. For each pair of events, we check event arguments agent_{sub} and agent_{obj} respectively. If none of the types for the aligned event arguments match, this pair is determined to be in conflict. If the event argument is missing, we deem it compatible with any type. In this procedure, we generate a set of event pairs Set_{conflict} that will not get co-reference links.

Given the event mention similarity as well as the conflicts, we perform event co-reference inference via a left-linking greedy algorithm, i.e. co-reference decisions are made on each event from left to right, one at a time. Without loss of generality, for event $e_{k+1}, \forall k \geq 1$, we first choose a linkable event to its left with the highest event-pair similarity:

$$e_p = \arg \max_{\substack{e \in \{e_1, e_2, \ldots, e_k\} \\ e \notin Set_{\text{conflict}}}} S(e, e_{k+1}).$$

We make co-reference links when $S(e_p, e_{k+1})$ is higher than a cut-off threshold, which is also tuned only on event examples ahead of time. Otherwise, event e_{k+1} is not similar enough to any of its antecedents, and we make it the start of a new cluster.

5.2 Experiments

5.2.1 Experimental Setup

Compared Systems

For baselines, **Joint** (Chen et al., 2009) is an early work based on supervised learning. We also report **HDP-Coref** results as an unsupervised baseline (Bejan and Harabagiu, 2010), which utilizes nonparametric Bayesian models. Moreover, we create another unsupervised event co-reference baseline (**Type+SharedMen**): we treat events of the same type which share at least one co-referent entity (inside event arguments) as co-referred. On TAC-KBP corpus, we report results from the top ranking system of the TAC-2015 Event Nugget Evaluation Task as **TAC-TOP**. Our developed similarity based co-reference detection **MSEP-Coref** method has a number of variations depending on the modular text-vector conversion method (**ESA, BC, W2V, DEP**), whether we use augmented ESA vector representation (**AUG**)[1], and whether we use knowledge during co-reference inference (**KNOW**). We also develop a supervised event co-reference system following the work of Sammons et al. (2015), namely $\mathbf{Supervised_{Base}}$. We also add additional event vector representations[2] as features to this supervised system and get $\mathbf{Supervised_{Extend}}$.

Evaluation Metrics

We use the widely recognized metrics MUC (Vilain et al., 1995), BCUB (Bagga and Baldwin, 1998), Entity-based CEAF (CEAF_e) (Luo, 2005) and report their average.

5.2.2 Empirical Results

The performance of different systems for event co-reference based on gold event triggers is shown in Table 5.1. The co-reference cut-off threshold is tuned by optimizing the CoNLL average score on ten selected ACE documents. The threshold is then fixed, thus we do not change it when evaluating on the TAC-KBP corpus. As we do cross-validation on ACE, we exclude these ten documents from test at all times. We regard this tuning procedure as "independent" and "ahead of time" because of the following reasons: 1) We could have used as threshold-tuning co-reference examples a few news documents from other sources; we just use

[1] It is only designed for ESA because the ESA vector for two concatenated text fragments is different from the sum of the ESA vectors of individual text fragments, unlike other methods.

[2] We add the best event vector representation empirically.

ACE (Cross-Validation)		MUC	B^3	$CEAF_e$	BLANC	AVG
Supervised	Graph	—	—	84.5	—	—
	Joint	74.8	92.2	87.0	—	—
	$Supervised_{Base}$	73.6	91.6	85.9	82.2	83.3
	$Supervised_{Extend}$	**74.9**	92.8	87.1	**83.8**	**84.7**
Unsupervised	Type+SharedMen	59.1	83.2	76.0	72.9	72.8
	HDP-Coref	—	83.8	76.7	—	—
MSEP	$MSEP\text{-}Coref_{ESA}$	65.9	91.5	85.3	81.8	81.1
	$MSEP\text{-}Coref_{BC}$	65.0	89.8	83.7	80.9	79.9
	$MSEP\text{-}Coref_{W2V}$	65.1	90.1	83.6	81.5	80.1
	$MSEP\text{-}Coref_{DEP}$	65.9	92.3	85.6	81.5	**81.3**
	$MSEP\text{-}Coref_{ESA+AUG}$	67.4	92.6	86.0	82.6	**82.2**
	$MSEP\text{-}Coref_{ESA+AUG+KNOW}$	68.0	**92.9**	**87.4**	83.2	**82.9**
	$MSEP\text{-}Coref_{ESA+AUG+KNOW}$ (GA)	68.8	92.5	87.7	83.4	83.1
TAC-KBP (Test Data)		MUC	B^3	$CEAF_e$	BLANC	AVG
Supervised	TAC-TOP	—	—	—	—	**75.7**
	$Supervised_{Base}$	63.8	83.8	75.8	74.0	74.4
	$Supervised_{Extend}$	**65.3**	84.7	76.8	**75.1**	**75.5**
Unsupervised	Type+SharedMen	56.4	77.5	69.6	68.7	68.1
MSEP	$MSEP\text{-}Coref_{ESA}$	57.7	83.9	76.9	72.9	72.9
	$MSEP\text{-}Coref_{BC}$	56.9	81.8	76.2	71.7	71.7
	$MSEP\text{-}Coref_{W2V}$	57.2	82.1	75.9	72.3	71.9
	$MSEP\text{-}Coref_{DEP}$	58.2	83.3	76.7	72.8	**72.8**
	$MSEP\text{-}Coref_{ESA+AUG}$	59.0	84.5	77.3	72.5	**73.3**
	$MSEP\text{-}Coref_{ESA+AUG+KNOW}$	59.9	**84.9**	**77.3**	73.1	**73.8**
	$MSEP\text{-}Coref_{ESA+AUG+KNOW}$ (GA)	60.5	84.0	77.7	73.5	73.9

Table 5.1: **Event Co-reference Results on Gold Event Triggers.** "$MSEP\text{-}Coref_{ESA,BC,W2V,DEP}$" are variations of the proposed MSEP event co-reference system using ESA, Brown Cluster, Word2Vec and Dependency Embedding representations respectively. "$MSEP\text{-}Coref_{ESA+AUG}$" uses augmented ESA event vector representation and "$MSEP\text{-}Coref_{ESA+AUG+KNOW}$" applies knowledge to detect conflicting events. (GA) means that we use gold event arguments instead of approximated ones from SRL.

ACE documents as a data source for simplicity. 2) We believe that the threshold only depends on event representation (the model) rather than data. 3) Tuning a single decision threshold is much cheaper than tuning a whole set of model parameters.

Results show that the proposed MSEP event co-reference system significantly outperforms baselines and achieves the same level of performance of supervised methods (82.9 v.s. 83.3 on ACE and 73.8 v.s. 74.4 on TAC-KBP). MSEP achieves better results on B^3 and $CEAF_e$ than supervised methods. Note that supervised methods usually generate millions of features (2.5M on ACE and 1.8M on TAC-KBP for $Supervised_{Base}$). In contrast, MSEP only has several thousands of non-zero dimensions in event representations. This means that our structured vector representations, through derived without explicit annotations, are far more expressive than traditional features. When we add the event vector representation (augmented ESA) as features in $Supervised_{Extend}$, we improve the overall performance by more than 1 point. When tested individually, DEP

performs the best among the four text-vector conversion methods while BC performs the worst. A likely reason is that BC has too few dimensions while DEP constructs the longest vector. However, the results show that our augmented ESA representation (Fig. 3) achieves even better results. When we use knowledge to detect conflicting events during inference, the system further improves. Note that event arguments for the proposed MSEP are predicted by SRL. We show that replacing them with gold event arguments, only slightly improves the overall performance, indicating that SRL arguments are robust enough for the event co-reference task.

Chapter 6

Semantic Language Models

Natural language understanding often requires deep semantic knowledge. Expanding on previous proposals, we suggest that some important aspects of semantic knowledge can be modeled as a language model if done at an appropriate level of abstraction. We develop two distinct models that capture semantic frame chains and discourse information while abstracting over the specific mentions of predicates and entities. For each model, we investigate four implementations: a "standard" N-gram language model and three discriminatively trained "neural" language models that generate embeddings for semantic frames. The quality of the semantic language models (SemLM) is evaluated both intrinsically, using perplexity and a narrative cloze test and extrinsically – we show that our SemLM helps improve performance on semantic natural language processing tasks such as co-reference resolution and discourse parsing.

6.1 Motivation

Natural language understanding often necessitates deep semantic knowledge. This knowledge needs to be captured at multiple levels, from words to phrases, to sentences, to larger units of discourse. At each level, capturing meaning frequently requires context sensitive abstraction and disambiguation, as shown in the following example (Winograd, 1972):

Ex.1 [Kevin] was **robbed** by [Robert]. [He] was **arrested** by the police.
Ex.2 [Kevin] was **robbed** by [Robert]. [He] was **rescued** by the police.

In both cases, one needs to resolve the pronoun "he" to either "Robert" or "Kevin". To make the correct decisions, one needs to know that the subject of "rob" is more likely than the object of "rob" to be the object of "arrest" while the object of "rob" is more likely to be the object of "rescue". Thus, beyond understanding individual predicates (e.g., at the semantic role labeling level), there is a need to place them and their arguments in a global context.

However, just modeling semantic frames is not sufficient; consider a variation of Ex.1:

Ex.3 Kevin was **robbed** by Robert, but the police mistakenly **arrested** him.

In this case, "him" should refer to "Kevin" as the discourse marker "but" reverses the meaning, illustrating that it is necessary to take discourse markers into account when modeling semantics.

In this chapter, we propose that these aspects of semantic knowledge can be modeled as a *Semantic Language Model* (SemLM). Just like the "standard" syntactic language models (LM), we define a basic vocabulary, a finite representation language, and a prediction task, which allows us to model the distribution over the occurrence of elements in the vocabulary as a function of their (well-defined) context. In difference from syntactic LMs, we represent natural language at a higher level of semantic abstraction, thus facilitating modeling deep semantic knowledge.

We propose two distinct discourse driven language models to capture semantics. In our first semantic language model, the *Frame-Chain SemLM*, we model all semantic frames and discourse markers in the text. Each document is viewed as a single chain of semantic frames and discourse markers. Moreover, while the vocabulary of discourse markers is rather small, the number of different surface form semantic frames that could appear in the text is very large. To achieve a better level of abstraction, we disambiguate semantic frames and map them to their PropBank/FrameNet representation. Thus, in Ex.3, the resulting frame chain is "rob.01 — but — arrest.01" ("01" indicates the predicate sense).

Our second semantic language model is called *Entity-Centered SemLM*. Here, we model a sequence of semantic frames and discourse markers involved in a specific co-reference chain. For each co-reference chain in a document, we first extract semantic frames corresponding to each co-referent mention, disambiguate them as before, and then determine the discourse markers between these frames. Thus, each unique frame contains both the disambiguated predicate and the argument label of the mention. In Ex.3, the resulting sequence is "rob.01#obj — but — arrest.01#obj" (here "obj" indicates the argument label for "Kevin" and "him" respectively). While these two models capture somewhat different semantic knowledge, we argue later that both models can be induced at high quality, and that they are suitable for different NLP tasks.

For both models of SemLM, we study four language model implementations: N-gram, skip-gram (Mikolov et al., 2013c), continuous bag-of-words (Mikolov et al., 2013b) and log-bilinear language model (Mnih and Hinton, 2007). Each model defines its own prediction task. In total, we produce eight different SemLMs. Except for N-gram model, others yield embeddings for semantic frames as they are neural language models.

In our empirical study, we evaluate both the quality of all SemLMs and their application to co-reference resolution and shallow discourse parsing tasks. Following the traditional evaluation standard of language models, we first use perplexity as our metric. We also follow the script learning literature (Chambers and

	F-Sen	F-Arg	Conn	Per	Seq/Doc
FC	YES	NO	YES	YES	Single
EC	YES	YES	YES	NO	Multiple

Table 6.1: Comparison of vocabularies between frame-chain (FC) and entity-centered (EC) SemLMs. "F-Sen" stands for frames with predicate sense information while "F-Arg" stands for frames with argument role label information; "Conn" means discourse marker and "Per" means period. "Seq/Doc" represents the number of sequence per document.

Jurafsky, 2008, 2009b; Rudinger et al., 2015) and evaluate on the narrative cloze test, i.e. randomly removing a token from a sequence and test the system's ability to recover it. We conduct both evaluations on two test sets: a hold-out dataset from the New York Times Corpus and gold sequence data (for frame-chain SemLMs, we use PropBank (Kingsbury and Palmer, 2002); for entity-centered SemLMs, we use Ontonotes (Hovy et al., 2006)). By comparing the results on these test sets, we show that we do not incur noticeable degradation when building SemLMs using preprocessing tools. Moreover, we show that SemLMs improves the performance of co-reference resolution, as well as that of predicting the sense of discourse connectives for both explicit and implicit ones.

The main contributions of our work can be summarized as follows:

1. The design of two novel discourse driven Semantic Language models, building on text abstraction and neural embeddings.

2. The implementation of high quality SemLMs that are shown to improve state-of-the-art NLP systems.

6.2 Two Basic Semantic Language Models

In this section, we describe how we capture sequential semantic information consisted of semantic frames and discourse markers as semantic units (i.e. the vocabulary).

6.2.1 Semantic Frames and Discourse Markers

Semantic Frames

A semantic frame is composed of a predicate and its corresponding argument participants. Here we require the predicate to be disambiguated to a specific sense, and we need a certain level of abstraction of arguments so that we can assign abstract labels. The design of PropBank frames (Kingsbury and Palmer, 2002) and FrameNet frames (Baker et al., 1998) perfectly fits our needs. They both have a limited set of frames (in the scale of thousands) and each frame can be uniquely represented by its predicate sense. These frames provide a good level of generalization as each frame can be instantiated into various surface forms in natural texts.

We use these frames as part of our vocabulary for SemLMs. Formally, we use the notation f to represent a frame. Also, we denote $\mathrm{fa} \triangleq \mathrm{f\#Arg}$ when referring to an argument role label (Arg) inside a frame (f).

Discourse Markers

We use discourse markers (connectives) to model discourse relationships between frames. There is only a limited number of unique discourse markers, such as *and, but, however*, etc. We get the full list from the Penn Discourse Treebank (Prasad et al., 2008) and include them as part of our vocabulary for SemLMs. Formally, we use *dis* to denote the discourse marker. Note that discourse relationships can exist without an explicit discourse marker, which is also a challenge for discourse parsing. Since we cannot reliably identify implicit discourse relationships, we only consider explicit ones here. More importantly, discourse markers are associated with arguments (Wellner and Pustejovsky, 2007) in text (usually two sentences/clauses, sometimes one). We only add a discourse marker in the semantic sequence when its corresponding arguments contain semantic frames which belong to the same semantic sequence. We call them *frame-related discourse markers*. We rely on a shallow discourse parsing tool (Song et al., 2015) to identify the explicit discourse markers and their corresponding arguments.

6.2.2 Frame-Chain SemLM

For frame-chain SemLM, we model all semantic frames and discourse markers in a document. We form the semantic sequence by first including all semantic frames in the order they appear in the text: $[\mathrm{f}_1, \mathrm{f}_2, \mathrm{f}_3, \ldots]$. Then we add *frame-related discourse markers* into the sequence by placing them in their order of appearance. Thus we get a sequence like $[\mathrm{f}_1, \mathrm{dis}_1, \mathrm{f}_2, \mathrm{f}_3, \mathrm{dis}_2, \ldots]$. Note that discourse markers do not necessarily exist between all semantic frames. Additionally, we treat the *period* symbol as a special discourse marker, denoted by "o". As some sentences contain more than one semantic frame (situations like clauses), we get the final semantic sequence like this:

$$[\mathrm{f}_1, \mathrm{dis}_1, \mathrm{f}_2, \mathrm{o}, \mathrm{f}_3, \mathrm{o}, \mathrm{dis}_2, \ldots, \mathrm{o}]$$

6.2.3 Entity-Centered SemLM

We generate semantic sequences according to co-reference chains for entity-centered SemLM. From co-reference resolution, we can get a sequence like $[\mathrm{m}_1, \mathrm{m}_2, \mathrm{m}_3, \ldots]$, where mentions appear in the order they occur in the text. Each mention can be matched to an argument inside a semantic frame. Thus, we replace each mention with its argument label inside a semantic frame, and get $[\mathrm{fa}_1, \mathrm{fa}_2, \mathrm{fa}_3, \ldots]$. We then add

discourse markers exactly in they way we do for frame-chain SemLM, and get the following sequence:

$$[\mathrm{fa}_1, \mathrm{dis}_1, \mathrm{fa}_2, \mathrm{fa}_3, \mathrm{dis}_2, \ldots]$$

The comparison of vocabularies between frame-chain and entity-centered SemLMs is summarized in Table 6.1.

6.3 Implementation of SemLMs

In this work, we experiment with four language model implementations: N-gram (NG), Skip-Gram (SG), Continuous Bag-of-Words (CBOW) and Log-bilinear (LB) language model. For ease of explanation, we assume that a semantic unit sequence is $s = [w_1, w_2, w_3, \ldots, w_k]$.

6.3.1 N-gram Model

For an n-gram model, we predict each token based on its $n-1$ previous tokens, i.e. we directly model the following conditional probability (in practice, we choose $n = 3$, Tri-gram (TRI)):

$$p(w_{t+2}|w_t, w_{t+1}).$$

Then, the probability of the sequence is

$$p(s) = p(w_1)p(w_2|w_1)\prod_{t=1}^{k-2} p(w_{t+2}|w_t, w_{t+1}).$$

To compute $p(w_2|w_1)$ and $p(w_1)$, we need to back off from Tri-gram to Bi-gram and Uni-gram.

6.3.2 Skip-Gram Model

The SG model was proposed in Mikolov et al. (2013c). It uses a token to predict its context, i.e. we model the following conditional probability:

$$p(c \in c(w_t)|w_t, \theta).$$

Here, $c(w_t)$ is the context for w_t and θ denotes the learned parameters which include neural network states and embeddings. Then the probability of the sequence is computed as

$$\prod_{t=1}^{k} \prod_{c \in c(w_t)} p(c|w_t, \theta).$$

6.3.3 Continuous Bag-of-Words Model

In contrast to skip-gram, CBOW (Mikolov et al., 2013b) uses context to predict each token, i.e. we model the following conditional probability:

$$p(w_t|c(w_t), \theta).$$

In this case, the probability of the sequence is

$$\prod_{t=1}^{k} p(w_t|c(w_t), \theta).$$

6.3.4 Log-bilinear Model

LB was introduced in Mnih and Hinton (2007). Similar to CBOW, it also uses context to predict each token. However, LB associates a token with three components instead of just one vector: a target vector v(w), a context vector v'(w) and a bias b(w). So, the conditional probability becomes:

$$p(w_t|c(w_t)) = \frac{\exp(v(w_t)^\intercal u(c(w_t)) + b(w_t))}{\sum_{w \in \mathcal{V}} \exp(v(w)^\intercal u(c(w_t)) + b(w))}.$$

Here, $\mathcal{V}$ denotes the vocabulary and we define $u(c(w_t)) = \sum_{c_i \in c(w_t)} q_i \odot v'(c_i)$. Note that $\odot$ represents element-wise multiplication and q_i is a vector that depends only on the position of a token in the context, which is a also a model parameter.

So, the overall sequence probability is

$$\prod_{t=1}^{k} p(w_t|c(w_t)).$$

6.4 Build SemLMs from Scratch

In this section, we explain how we build SemLMs from un-annotated plain text.

6.4.1 Dataset and Preprocessing

Dataset

We use the New York Times Corpus[1] (from year 1987 to 2007) for training. It contains a bit more than 1.8M documents in total.

Preprocessing

We pre-process all documents with semantic role labeling (Punyakanok et al., 2004) and part-of-speech tagger (Roth and Zelenko, 1998). We also implement the explicit discourse connective identification module in shallow discourse parsing (Song et al., 2015). Additionally, we utilize within document entity co-reference (Peng et al., 2015a) to produce co-reference chains. To obtain all annotations, we employ the Illinois NLP tools[2].

6.4.2 Semantic Unit Generation

FrameNet Mapping

We first directly derive semantic frames from semantic role labeling annotations. As the Illinois SRL package is built upon PropBank frames, we do a mapping to FrameNet frames via VerbNet senses (Schuler, 2005), thus achieving a higher level of abstraction. The mapping file[3] defines deterministic mappings. However, the mapping is not complete and there are remaining PropBank frames. Thus, the generated vocabulary for SemLMs contains both PropBank and FrameNet frames. For example, "place" and "put" with the VerbNet sense id "9.1-2" are converted to the same FrameNet frame "Placing".

Augmenting to Verb Phrases

We apply three heuristic modifications to augment semantic frames: 1) if a preposition immediately follows a predicate, we append the preposition to the predicate e.g. "take over"; 2) if we encounter the semantic role label AM-PRD which indicates a secondary predicate, we also append this secondary predicate to the main predicate e.g. "be happy"; 3) if we see the semantic role label AM-NEG which indicates negation, we append "not" to the predicate e.g. "not like". These three augmentations can co-exist and they allow us to model more fine-grained semantic frames.

Verb Compounds

We have observed that if two predicates appear very close to each other, e.g. "eat and drink", "decide to buy", they actually represent a unified semantic meaning. Thus, we construct compound verbs to connect them together. We apply the rule that if the gap between two predicates is less than two tokens, we treat

[1] https://catalog.ldc.upenn.edu/LDC2008T19
[2] http://cogcomp.cs.illinois.edu/page/software/
[3] http://verbs.colorado.edu/verb-index/fn/vn-fn.xml

	Vocabulary Size			Sequence Size		
	F-s	F-c	Conn	#seq	#token	#t/s
FC	14857	7269	44	1.2M	25.4M	21
EC	8758	2896	44	3.4M	18.6M	5
LM	∼20k			∼3M	∼38M	10-15

Table 6.2: **Statistics on SemLM vocabularies and sequences.** "F-s" stands for single frame while "F-c" stands for compound frame; "Conn" means discourse marker. "#seq" is the number of sequences, and "#token" is the total number of tokens (semantic units). We also compute the average token in a sequence i.e. "#t/s". We compare frame-chain (FC) and entity-centered (EC) SemLMs to the usual syntactic language model setting i.e. "LM".

them as a unified semantic frame defined by the conjunction of the two (augmented) semantic frames, e.g. "eat.01-drink.01" and "decide.01-buy.01".

Argument Labels for Co-referent Mentions

To get the argument role label information for co-referent mentions, we need to match each mention to its corresponding semantic role labeling argument. If a mention head is inside an argument, we regard it as a match. We do not consider singleton mentions.

Vocabulary Construction

After generating all semantic units for (augmented and compounded) semantic frames and discourse markers, we merge them together as a tentative vocabulary. In order to generate a sensible SemLM, we filter out rare tokens which appear less than 20 times in the data. We add the Unknown token (UNK) and End-of-Sequence token (EOS) to the eventual vocabulary.

Statistics on the eventual SemLM vocabularies and semantic sequences are shown in Table 6.2. We also compare frame-chain and entity-centered SemLMs to the usual syntactic language model setting. The statistics in Table 6.2 shows that they are comparable both in vocabulary size and in the total number of tokens for training. Moreover, entity-centered SemLMs have shorter sequences then frame-chain SemLMs. We also provide several examples of high-frequency augmented compound semantic frames in our generated SemLM vocabularies. All are very intuitive:

want.01-know.01, agree.01-pay.01, try.01-get.01, decline.02-comment.01,
wait.01-see.01, make.02-feel.01, want.01(not)-give.08(up)

6.4.3 Language Model Training

NG

We implement the N-gram model using the SRILM toolkit (Stolcke, 2002). We also employ the well-known

Kneser–Ney Smoothing (Kneser and Ney, 1995) technique.

SG & CBOW

We utilize the word2vec package to implement both SG and CBOW. In practice, we set the context window size to be 10 for SG while set the number as 5 for CBOW (both are usual settings for syntactic language models). We generate 300-dimension embeddings for both models.

LB

We use the OxLM toolkit (Paul et al., 2014) with Noise-Constrastive Estimation (Gutmann and Hyvarinen, 2010) for the LB model. We set the context window size to 5 and produce 150-dimension embeddings.

6.5 Evaluations

In this section, we first evaluate the quality of SemLMs through perplexity and a narrative cloze test. More importantly, we show that the proposed SemLMs can help improve the performance of co-reference resolution and shallow discourse parsing. This further proves that we successfully capture semantic sequence information which can potentially benefit a wide range of semantic related NLP tasks.

We have designed two models for SemLM: *frame-chain* (**FC**) and *entity-centered* (**EC**). By training on both types of sequences respectively, we implement four different language models: **TRI, SG, CBOW, LB**. We focus the evaluation efforts on these eight SemLMs.

6.5.1 Quality Evaluation of SemLMs

Datasets

We use three datasets. We first randomly sample 10% of the New York Times Corpus documents (roughly two years of data), denoted the *NYT Hold-out Data*. All our SemLMs are trained on the remaining NYT data and tested on this hold-out data. We generate semantic sequences for the training and test data using the methodology described in Sec. 6.4.2.

We use PropBank data with gold frame annotations as another test set. In this case, we only generate frame-chain SemLM sequences by applying semantic unit generation techniques on gold frames. When we test on *Gold PropBank Data with Frame Chains*, we use frame-chain SemLMs trained from all NYT data.

Similarly, we use Ontonotes data (Hovy et al., 2006) with gold frame and co-reference annotations as the third test set, *Gold Ontonotes Data with Coref Chains*. We only generate entity-centered SemLMs by applying semantic unit generation techniques on gold frames and gold co-reference chains.

Baselines

	Baselines		SemLMs			
	UNI	BG	TRI	CBOW	SG	LB
NYT Hold-out Data						
FC	952.1	178.3	119.2	115.4	114.1	**108.5**
EC	914.7	154.4	114.9	111.8	113.8	**109.7**
Gold PropBank Data with Frame Chains						
FC-FM	992.9	213.7	139.1	135.6	128.4	121.8
FC	970.0	191.2	132.7	126.4	123.5	**115.4**
Gold Ontonotes Data with Coref Chains						
EC-FM	956.4	187.7	121.1	115.6	117.2	113.7
EC	923.8	163.2	120.5	113.7	115.0	**109.3**

Table 6.3: Perplexities for SemLMs. UNI, BG, TRI, CBOW, SG, LB are different language model implementations while "FC" and "EC" stand for the two SemLM models studied, respectively. "FC-FM" and "EC-FM" indicate that we removed the "FrameNet Mapping" step. Note that for CBOW, SG and LB models, the perplexity numbers are not directly comparable with the N-gram model.

We use Uni-gram (**UNI**) and Bi-gram (**BG**) as two language model baselines. In addition, we use the point-wise mutual information (PMI) for token prediction. Essentially, PMI scores each pair of tokens according to their co-occurrences. It predicts a token in the sequence by choosing the one with the highest total PMI with all other tokens in the sequence. We use the ordered PMI (**OP**) as our baseline, which is a variation of PMI by considering asymmetric counting (Jans et al., 2012).

Perplexity Results

As SemLMs are language models, it is natural to evaluate the perplexity, which is a measurement of how well a language model can predict sequences. Results for SemLM perplexities are presented in Table 6.3. They are computed without considering end token (EOS). Note that for CBOW, SG and LB models, the perplexity numbers are not directly comparable with the N-gram model. For the three neural models, we compute the probabilities of each semantic unit from the soft-max layer, and averaged them over the length of the sequence. Similar to syntactic language models, perplexities are fast decreasing from UNI, BI to TRI.

We can compare the results of our frame-chain SemLM on *NYT Hold-out Data* and *Gold PropBank Data with Frame Chains*, and our entity-centered SemLM on *NYT Hold-out Data* and *Gold Ontonotes Data with Coref Chains*. While we see differences in the results, the gap is narrow. This indicates that the automatic SRL and Co-reference annotations added some noise but, more importantly, that the resulting SemLMs are robust to this noise as we still retain the language modeling ability for all methods. Additionally, our ablation study removes the "FrameNet Mapping" step ("FC-FM" and "EC-FM" rows), resulting in only using PropBank frames in the vocabulary. The increase in perplexities shows that "FrameNet Mapping" does produce a higher level of abstraction, which is useful for language modeling.

Narrative Cloze Test

	Baselines			SemLMs				Rel-Impr
	OP	UNI	BG	TRI	CBOW	SG	LB	
				MRR				
NYT Hold-out Data								
FC	0.121	0.236	0.225	0.249	0.242	0.247	**0.276**	8.5%
EC	0.126	0.235	0.210	0.242	0.249	0.249	**0.261**	5.9%
EC w/o DIS	0.092	0.191	0.188	0.212	0.215	0.216	**0.227**	18.8%
Rudinger et al. (2015)*	0.083	0.186	0.181	—	—	—	**0.223**	19.9%
Gold PropBank Data with Frame Chains								
FC	0.106	0.215	0.212	0.232	0.228	0.229	**0.254**	18.1%
FC-FM	0.098	0.201	0.204	0.223	0.218	0.220	0.243	—
Gold Ontonotes Data with Coref Chains								
EC	0.122	0.228	0.213	0.239	0.247	0.246	**0.257**	12.7%
EC-FM	0.109	0.215	0.208	0.230	0.237	0.239	0.254	—
				Recall@30				
NYT Hold-out Data								
FC	33.2	46.8	45.3	47.3	46.6	47.5	**55.4**	18.4%
EC	29.4	43.7	41.6	44.8	46.5	46.6	**52.0**	19.0%
Gold PropBank Data with Frame Chains								
FC	26.3	39.5	38.1	45.5	43.6	43.8	**53.9**	36.5%
FC-FM	24.4	37.3	37.3	42.8	41.9	42.1	48.2	—
Gold Ontonotes Data with Coref Chains								
EC	30.6	42.1	39.7	46.4	48.3	48.1	**51.5**	22.3%
EC-FM	26.6	39.9	37.6	45.4	46.7	46.2	49.8	—

Table 6.4: Narrative cloze test results for SemLMs. UNI, BG, TRI, CBOW, SG, LB are different language model implementations while "FC" and "EC" stand for our two SemLM models, respectively. "FC-FM" and "EC-FM" mean that we remove the FrameNet mappings. "w/o DIS" indicates the removal of discourse makers in SemLMs. "Rel-Impr" indicates the relative improvement of the best performing SemLM over the strongest baseline. We evaluate on two metrics: mean reciprocal rank (MRR)/recall at 30 (Recall@30). LB outperforms other methods for both frame-chain and entity-centered SemLMs.

We follow the Narrative Cloze Test idea used in script learning (Chambers and Jurafsky, 2008, 2009b). As Rudinger et al. (2015) points out, the narrative cloze test can be regarded as a language modeling evaluation. In the narrative cloze test, we randomly choose and remove one token from each semantic sequence in the test set. We then use language models to predict the missing token and evaluate the correctness. For all SemLMs, we use the conditional probabilities to get token predictions. We also use ordered PMI as an additional baseline. The narrative cloze test is conducted on the same test sets as the perplexity evaluation. We use mean reciprocal rank (MRR) and recall at 30 (Recall@30) to evaluate.

Results are provided in Table 6.4. LB outperforms other methods for both frame-chain and entity-centered SemLMs across all test sets. It is interesting to see that UNI performs better than BG in this prediction task. This finding is also reflected in the results reported in Rudinger et al. (2015). With respect to the strongest baseline (UNI), LB achieves close to 20% relative improvement for Recall@30 metric on NYT hold-out data. On gold data, the frame-chain SemLMs get a relative improvement of 36.5% for Recall@30 while

	ACE04	CoNLL12
Wiseman et al. (2015)	—	63.39
Base (Peng et al., 2015a)	71.20	63.03
Base+EC-TRI (p_c)	71.31	63.14
Base+EC-TRI w/o DIS	71.08	62.99
Base+EC-LB (p_c)	71.71	63.42
Base+EC-LB ($p_c + em$)	**71.79**	**63.46**
Base+EC-LB w/o DIS	71.12	63.00
Wiseman et al. (2016)	—	64.2
Clark and Manning (2016a)	—	65.7
Lee et al. (2017)	—	68.8
Peters et al. (2018)	—	**70.4**

Table 6.5: Co-reference resolution results with entity-centered SemLM features. "EC" stands for the entity-centered SemLM. "TRI" is the tri-gram model while "LB" is the log-bilinear model. "p_c" means conditional probability features and "em" represents frame embedding features. "w/o DIS" indicates the ablation study by removing all discourse makers for SemLMs. We conduct the experiments by adding SemLM features into the base system. We outperform the state-of-art system (Wiseman et al., 2015) (as in 2015). The improvement achieved by "EC_LB ($p_c + em$)" over the base system is statistically significant. We also include more recent results where the current state-of-the-art performance is achieved by Peters et al. (2018) using ELMo embeddings.

entity-centered SemLMs get 22.3%. For MRR metric, the relative improvement is around half that of the Recall@30 metric. In the narrative cloze test, we also carry out an ablation study to remove the "FrameNet Mapping" step ("FC-FM" and "EC-FM" rows). The decrease in MRR and Recall@30 metrics further strengthens the argument that "FrameNet Mapping" is important for language modeling as it improves the generalization on frames.

We cannot directly compare with other related works (Rudinger et al., 2015; Pichotta and Mooney, 2016) because of the differences in data and evaluation metrics. Rudinger et al. (2015) also use the NYT portion of the Gigaword corpus, but with Concrete annotations; Pichotta and Mooney (2016) use the English Wikipedia as their data, and Stanford NLP tools for pre-processing while we use the Illinois NLP tools. Consequently, the eventual chain statistics are different, which leads to different test instances.[4] We counter this difficulty by reporting results on "Gold PropBank Data" and "Gold Ontonotes Data". We hope that these two gold annotation datasets can become standard test sets. Rudinger et al. (2015) does share a common evaluation metric with us: MRR. If we ignore the data difference and make a rough comparison, we find that the absolute values of our results are better while Rudinger et al. (2015) have higher relative improvement ("Rel-Impr" in Table 4). This means that 1) the discourse information is very likely to help better model semantics 2) the discourse information may boost the baseline (UNI) more than it does for the LB model.

[4]Rudinger et al. (2015) is similar to our entity-centered SemLM without discourse information. So, in Table 4, we make a rough comparison between them.

	CoNLL16 Test			CoNLL16 Blind		
	Explicit	Implicit	Overall	Explicit	Implicit	Overall
Base (Song et al., 2015)	89.8	35.6	60.4	75.8	31.9	52.3
Base + FC-TRI (q_c)	90.3	35.8	60.7	76.4	32.5	52.9
Base + FC-TRI w/o DIS	89.2	35.3	60.0	75.5	31.6	52.0
Base + FC-LB (q_c)	90.9	36.2	61.3	76.8	32.9	53.4
Base + FC-LB ($q_c + em$)	**91.1**	**36.3**	**61.4**	**77.3**	**33.2**	**53.8**
Base + FC-LB w/o DIS	90.1	35.7	60.6	76.9	33.0	53.5

Table 6.6: Shallow discourse parsing results with frame-chain SemLM features. "FC" stands for the frame-chain SemLM. "TRI" is the tri-gram model while "LB" is the log-bilinear model. "p_c", "em" are conditional probability and frame embedding features, resp. "w/o DIS" indicates the case where we remove all discourse makers for SemLMs. We do the experiments by adding SemLM features to the base system. The improvement achieved by "FC-LB ($p_c + em$)" over the baseline is statistically significant.

6.5.2 Evaluation of SemLM Applications

Entity Co-reference

Co-reference resolution is the task of identifying mentions that refer to the same entity. To help improve its performance, we incorporate SemLM information as features into an existing co-reference resolution system. We choose the state-of-art Illinois Co-reference Resolution system (Peng et al., 2015a) as our base system. It employs a supervised joint mention detection and co-reference framework. We add additional features into the mention-pair feature set.

Given a pair of mentions (m_1, m_2) where m_1 appears before m_2, we first extract the corresponding semantic frame and the argument role label of each mention. We do this by following the procedures in Sec. 6.4.2. Thus, we can get a pair of semantic frames with argument information $(\text{fa}_1, \text{fa}_2)$. We may also get an additional discourse marker between these two frames, e.g. $(\text{fa}_1, \text{dis}, \text{fa}_2)$. Now, we add the following conditional probability as the feature from SemLMs:

$$p_c = p(\text{fa}_2|\text{fa}_1, \text{dis}).$$

We also add $p_c^2, \sqrt{p_c}$ and $1/p_c$ as features. To get the value of p_c, we follow the definitions in Sec. 6.2.1, and we only use the entity-centered SemLM here as its vocabulary covers frames with argument labels. For the neural language model implementations (CBOW, SG and LB), we also include frame embeddings as additional features.

We evaluate the effect of the added SemLM features on two co-reference benchmark datasets: ACE04 (NIST, 2004) and CoNLL12 (Pradhan et al., 2012). We use the standard split of 268 training documents, 68 development documents, and 106 testing documents for ACE04 data (Culotta et al., 2007; Bengtson and Roth, 2008). For CoNLL12 data, we follow the train and test document split from CoNLL-2012 Shared Task. We

report CoNLL AVG for results (average of MUC, B^3, and $CEAF_e$ metrics), using the v7.0 scorer provided by the CoNLL-2012 Shared Task.

Co-reference resolution results with entity-centered SemLM features are shown in Table 6.5. Tri-grams with conditional probability features improve the performance by a small margin, while the log-bilinear model achieves a 0.4-0.5 F1 points improvement. By employing log-bilinear model embeddings, we further improve the numbers and we outperform the best reported results on the CoNLL12 dataset (Wiseman et al., 2015) (as in 2015). We also include more recent results where the current state-of-the-art performance is achieved by Peters et al. (2018) using ELMo embeddings.

In addition, we carry out ablation studies to remove all discourse makers during the language modeling process. We re-train our models and study their effects on the generated features. Table 6.5 ("w/o DIS" rows) shows that without discourse information, the SemLM features would hurt the overall performance, thus proving the necessity of considering discourse for semantic language models.

Shallow Discourse Parsing

Shallow discourse parsing is the task of identifying explicit and implicit discourse connectives, determine their senses and their discourse arguments. In order to show that SemLM can help improve shallow discourse parsing, we evaluate on identifying the correct sense of discourse connectives (both explicit and implicit ones).

We choose Song et al. (2015), which uses a supervised pipeline approach, as our base system. The system extracts context features for potential discourse connectives and applies the discourse connective sense classifier. Consider an explicit connective "dis"; we extract the semantic frames that are closest to it (left and right), resulting in the sequence [f_1, dis, f_2] by following the procedures described in Sec. 6.4.2. We then add the following conditional probabilities as features. Compute

$$q_c = p(\text{dis}|\text{f}_1, \text{f}_2).$$

and, similar to what we do for co-reference resolution, we add $q_c, q_c^2, \sqrt{q_c}, 1/q_c$ as conditional probability features, which can be computed following the definitions in Sec. 6.2.1. We also include frame embeddings as additional features. We only use frame-chain SemLMs here.

We evaluate on CoNLL16 (Xue et al., 2016) test and blind sets, following the train and development document split from the Shared Task, and report F1 using the official shared task scorer.

Table 6.6 shows the results for shallow discourse parsing with SemLM features. Tri-gram with conditional probability features improve the performance for both explicit and implicit connective sense classifiers. Log-

bilinear model with conditional probability features achieves even better results, and frame embeddings further improve the numbers. SemLMs improve relatively more on explicit connectives than on implicit ones.

We also show an ablation study in the same setting as we did for co-reference, i.e. removing discourse information ("w/o DIS" rows). While our LB model can still exhibit improvement over the base system, its performance is lower than the proposed discourse driven version, which means that discourse information improves the expressiveness of semantic language models.

Chapter 7

SemLM with Multiple Semantic Aspects

In this chapter, We augment the developed semantic language models with more semantic aspects.

7.1 Motivation

Understanding a story requires understanding sequences of events. It is thus vital to model semantic sequences in text. This modeling process necessitates deep semantic knowledge about what can happen next. Since events involve actions, participants and emotions, semantic knowledge about these aspects must be captured and modeled.

Models	Context Input	Generated Ending
4-gram	Steven Avery committed murder. He was arrested, charged and tried.	With law by the judge <UNK> ...
RNNLM	*same as above*	The information under terrorism ...
Seq2Seq	*same as above*	He decided for a case.
FC-SemLM	commit.01 arrest.01 charge.05 try.01	convict.01
FES-LM	PER[new]-commit.01-ARG[new](NEG) ARG[new]-arrest.01-PER[old](NEU) ARG[new]-charge.05-PER[old](NEU) ARG[new]-try.01-PER[old](NEG)	ARG[new]-convict.01-PER[old](NEG)

Table 7.1: Comparison of generative ability for different models. For each model, we provide Ex.1 as context and compare the generated ending. 4-gram and RNNLM models are trained on NYT news data while Seq2Seq model is trained on the story data (details see Sec. 7.5). These are models operated on the word level. We compare them with FC-SemLM (Peng and Roth, 2016), which works on frame abstractions, i.e. "predicate.sense". For the proposed FES-LM, we further assign the arguments (subject and object) of a predicate with NER types ("PER, LOC, ORG, MISC") or "ARG" if otherwise. Each argument is also associated with a "[new/old]" label indicating if it is first mentioned in the sequence (decided by entity co-reference). Additionally, the sentiment of a frame is represented as positive (POS), neural (NEU) or negative (NEG). FES-LM can generate better endings in terms of soundness and specificity. The FES-LM ending can be understood as "[Something] convict a person, who has been mentioned before (with an overall negative sentiment)", which can be instantiated as "Steven Avery was convicted." given current context.

Consider the examples in Figure 7.1. In Ex.1, we observe a sequence of actions (commit, arrest, charge, try), each corresponding to a predicate frame. Clearly, "convict" is more likely than "go" to follow such sequence. This semantic knowledge can be learned through modeling frame sequences observed in a large

Ex.1 (Actions - Frames) Steven Avery *committed* murder. He was *arrested*, *charged* and *tried*.

Opt.1 Steven Avery was convicted of murder.
Opt.2 Steven went to the movies with friends.
Alter. Steven was held in jail during his trial.

Ex.2 (Participants - Entities) It was my first time ever playing *football* and I was so nervous. During the game, I got tackled and it did not hurt at all!

Opt.1 I then felt more confident playing football.
Opt.2 I realized playing baseball was a lot of fun.
Alter. However, I still love baseball more.

Ex.3 (Emotions - Sentiments) Joe wanted to become a professional plumber. So, he applied to a trade school. Fortunately, he was *accepted*.

Opt.1 It made Joe very happy.
Opt.2 It made Joe very sad.
Alter. However, Joe decided not to enroll because he did not have enough money to pay tuition.

Figure 7.1: Examples of short stories requiring different aspects of semantic knowledge. For all stories, Opt.1 is the correct follow-up, while Opt.2 is the contrastive wrong follow-up demonstrating the importance of each aspect. Alter. showcases an alternative correct follow-up, which requires considering different aspects of semantics jointly.

corpus. This phenomena has already been studied in script learning works (Chatman, 1980; Chambers and Jurafsky, 2008; Ferraro and Van Durme, 2016; Pichotta and Mooney, 2016; Peng and Roth, 2016). However, modeling actions is not sufficient; participants in actions and their emotions are also important. In Ex. 2, Opt.2 is not a plausible answer because the story is about "football", and it does not make sense to suddenly change the key entity to "baseball". In Ex.3, one needs understand that "being accepted" typically indicates a positive sentiment and that it applies to "Joe".

As importantly, we believe that modeling these semantic aspects should be done jointly; otherwise, it may not convey the complete intended meaning. Consider the alternative follow-ups in Figure 7.1: in Ex.1, the entity "jail" gives strong indication that it follows the storyline that mentions "murder"; in Ex.2, even though "football" is not explicitly mentioned, there is a comparison between "baseball" and "football" that makes this continuation coherent; in Ex.3, "decided not to enroll" is a reasonable action after "being accepted", although the general sentiment of the sentence is negative. These examples show that in order to model semantics in a more complete way, we need to consider interactions between frames, entities and sentiments.

In this thesis, we propose a joint semantic language model, FES-LM, for semantic sequences, which captures **F**rames, **E**ntities and **S**entiment information. Just as "standard" language models built on top of words, we construct FES-LM by building language models on top of joint semantic representations. This joint semantic representation is a mixture of representations corresponding to different semantic aspects. For each aspect, we capture semantics via abstracting over and disambiguating text surface forms, i.e. semantic

frames for predicates, entity types for semantic arguments, and sentiment labels for the overall context. These abstractions provide the basic vocabulary for FES-LM and are essential for capturing the underlying semantics of a story. In Table 7.1, we provide Ex.1 as context input (although FC-SemLM and FES-LM automatically generate a more abstract representation of this input) and examine the ability of different models to generate an ending. 4-gram, RNNLM and Seq2Seq models operate on the word level, and the generated endings are not satisfactory. FC-SemLM (Peng and Roth, 2016) works on basic frame abstractions and the proposed FES-LM model adds abstracted entity and sentiment information into frames. The results show that FES-LM produces the best ending among all compared models in terms of semantic soundness and specificity.

We build the joint language model from plain text corpus with automatic annotation tools, requiring no human effort. In the empirical study, FES-LM is first built on news documents. We provide narrative cloze test results for different variants of FES-LM, where we test the system's ability to recover a randomly dropped frame. We further show that FES-LM improves the performance of sense disambiguation for shallow discourse parsing. We then re-train the model on short commonsense stories (with the model trained on news as initialization). We perform story cloze test (Mostafazadeh et al., 2017), i.e. given a four-sentence story, choose the fifth sentence from two provided options. Our joint model achieves the best known results in the unsupervised setting. In all cases, our ablation study demonstrates that each aspect of FES-LM contributes to the model.

The main contributions of our work are:

1. The design of a joint neural language model for semantic sequences built from frames, entities and sentiments.

2. Showing that FES-LM trained on news is of high quality and can help to improve shallow discourse parsing.

3. Achieving the state-of-the-art result on story cloze test in an unsupervised setting with the FES-LM tuned on stories.

7.2 Semantic Aspect Modeling

This section describes how we capture different aspects of the semantic information in a text snippet via semantic frames, entities and sentiments.

Ex.4 The doctor told Susan that *she* was busy.
The doctor told Susan that *she* had cancer.
Mary told Susan that *she* had cancer.

Figure 7.2: Examples of the need for different levels of entity abstraction. For each sentence, one wants to understand what the pronoun "she" refers to, which requires different abstractions for two underlined entity choices depending on context.

7.2.1 Semantic Frames

Semantic frame is defined by Fillmore (1976): *frames are certain schemata or frameworks of concepts or terms which link together as a system, which impose structure or coherence on some aspect of human experience, and which may contain elements which are simultaneously parts of other such frameworks.* In this work, we simplify it by defining a semantic frame as a composition of a predicate and its corresponding argument participants. The design of PropBank frames (Kingsbury and Palmer, 2002) and FrameNet frames (Baker et al., 1998) perfectly fits our needs. Here we require the predicate to be disambiguated to a specific sense, thus each frame can be uniquely represented by its predicate sense. These frames provide a good level of generalization as each frame can be instantiated into various surface forms in natural texts. For example, in Ex.1, the semantic frame in Opt.1 would be abstracted as "convict.01". We associate each of these frames with an embedding. The arguments of the frames are modeled as entities, as described next.

Additionally, in accordance with the idea proposed by Peng and Roth (2016), we also extend the frame representations to include discourse markers since they model relationships between frames. In this work, we only consider explicit discourse markers between abstracted frames. We use surface forms to represent discourse markers because there is only a limited set. We also assign an embedding with the same dimension as frames to each discourse marker.

To unify the representation, we formally use e_f to represent an embedding of a disambiguated frame/discourse marker. Such embedding would later be learned during language model training.

7.2.2 Entities

We consider the subject and object of a predicate as the essential entity information for modeling semantics. To achieve a higher level of abstraction, we model entity types instead of entity surface forms. We choose to assign entities with labels produced by Named Entity Recognition (NER), as NER typing is reliable.[1]

In fact, it is difficult to abstract each entity into an appropriate level since the decision is largely affected by context. Consider the examples shown in Figure 7.2. For the first sentence, to correctly understand

[1]Though there are a number works on fine-grained entity typing (Yogatama et al., 2015; Ren et al., 2016), their performances are between 65% and 75%, much lower than NER.

what "she" refers to, it is enough to just abstract both entities "the doctor" and "Susan" to the NER type "person", i.e. the semantic knowledge being *person A told person B that person A was busy.* However, when we change the context in the second sentence, the "person" abstraction becomes too broad as it loses key information for this "doctor - patient" situation. The ideal semantic abstraction would be "a doctor told a patient that the patient had a disease". For the third sentence, it is ambiguous without further context from other sentences. Thus, entity abstraction is a delicate balance between specificity and correctness.

Besides type information, Ex.2 in Figure 7.1 shows the necessity of providing *new entity* information, i.e whether or not an entity is appeared for the first time in the whole semantic sequence. This corresponds well with the definition of *anaphrocity* in co-reference resolution, i.e. whether or not the mention starts a co-reference chain. Thus, we can encode this binary information as an additional dimension in the entity representation.

Thus, we formally define r_e as the entity representation. It is the concatenation of two entity vectors r_{sub} and r_{obj} for subject entity and object entity respectively. Both r_{sub} and r_{obj} are constructed as a one hot vector[2] to represent an entity type, plus an additional dimension indicating whether or not it is a *new entity* (1 if it is new).

7.2.3 Sentiments

For a piece of text, we can assign a sentiment value to it. It can either be positive, negative, or neutral. In order to decide which one is most appropriate, we first use a look-up table from word lexicons to sentiment, and then count the number of words which corresponds to positive (n_{pos}) and negative (n_{neg}) sentiment respectively. If $n_{pos} > n_{neg}$, we determine the text as positive; and if $n_{pos} < n_{neg}$, we assign the negative label; and if the two numbers equal, we deem the text as neutral. We use one hot vector for three sentiment choices, and define sentiment representation as r_s.

7.3 FES-LM - Joint Modeling

We present our joint model FES-LM and the neural language model implementation in this section. The joint model considers frames, entities and sentiments together to construct FES representations in order to model semantics more completely. Moreover, we build language models on top of such representations to reflect the sequential nature of semantics.

[2]Each dimension of the vector indicates an entity type (binary 0/1), and the vector contains exactly one element of 1.

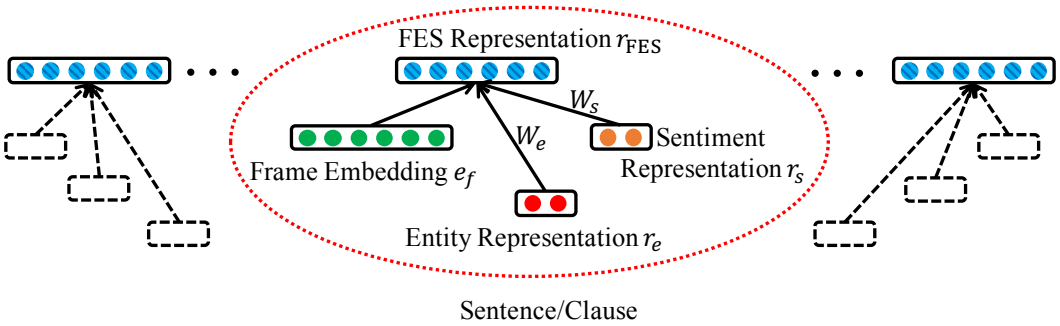

Figure 7.3: An overview of the FES representation in a semantic sequence. Semantic frames are represented by vector r_f. The entity representation r_e is the concatenation of r_{sub} and r_{obj}, both consist of two parts: an one-hot vector for entity type plus an additional dimension to indicate whether or not it is a *new entity*. The sentiment representation r_s is also one-hot.

7.3.1 FES Representation

We propose FES-LM as a joint model to embed frame, entity and sentiment information together. Thus for each sentence/clause (specific to a frame), we can get individual representations for the frame (i.e. e_f), entity types and new entity information corresponds to subject and object of the frame (i.e. r_e), and sentiment information (i.e. r_s). Thus, we construct the FES representation as:

$$r_{\text{FES}} = e_f + W_e r_e + W_s r_s.$$

W_e, W_s are two matrices transforming entity and sentiment representations into the frame embedding space, which are added to the corresponding frame embedding. These two parameters are shared across all FES representations. During language model training, we learn frame embeddings e_f as well as W_e and W_s. An overview of the FES representation in a semantic sequence is shown in Figure 7.3. Note that if the frame embedding represents a discourse marker, we set the corresponding entity and sentiment representations as zero vectors since no entity/sentiment is matched to a discourse marker. It is our design choice to add the entity and sentiment vectors to the frame embeddings, which creates a unified semantic space. During training, the interactions between different semantic aspects are captured by optimizing the loss on the joint FES representations.[3]

7.3.2 Neural Language Model

To model semantic sequences and train FES representations, we build neural language models. Theoretically, we can utilize any existing neural language model. We choose to implement the log-bilinear language model

[3]An alternative design choice is to concatenate the vector representations from different semantic aspects together, but we did not get better empirical results compared to our current design.

(LBL) (Mnih and Hinton, 2007) as our main method since previous works have reported best performance using it (Rudinger et al., 2015; Peng and Roth, 2016).

For ease of explanation, we assume that a semantic sequence of FES representations is $[\text{FES}_1, \text{FES}_2, \text{FES}_3, \ldots, \text{FES}_k]$, with FES_i being the i_{th} FES representation in the sequence. It assigns each token (i.e. FES representation) with three components: a target vector $v(\text{FES})$, a context vector $v'(\text{FES})$ and a bias $b(\text{FES})$. Thus, we model the conditional probability of a token FES_t given its context $c(\text{FES}_t)$:

$$p(\text{FES}_t|c(\text{FES}_t)) = \frac{\exp(v(\text{FES}_t)^\intercal u(c(\text{FES}_t)) + b(\text{FES}_t))}{\sum_{\text{FES}\in\mathcal{V}} \exp(v(\text{FES})^\intercal u(c(\text{FES}_t)) + b(\text{FES}))}.$$

Here, $\mathcal{V}$ denotes the vocabulary (all possible FES representations) and we define

$$u(c(\text{FES}_t)) = \sum_{c_i \in c(\text{FES}_t)} q_i \odot v'(c_i).$$

Note that $\odot$ represents element-wise multiplication and q_i is a vector that depends only on the position of an FES representation in context, which is also a model parameter. For language model training, we maximize the overall sequence probability $\prod_{t=1}^{k} p(\text{FES}_t|c(\text{FES}_t))$.

7.4 Building FES-LM

In this section, we explain how we build FES-LM from un-annotated plain text.

7.4.1 Dataset and Preprocessing

Dataset

We first use the New York Times (NYT) Corpus[4] (from year 1987 to 2007) to train FES-LM. It contains over 1.8M documents in total. To fine tune the model on short stories, we re-train FES-LM on the ROCStories dataset (Mostafazadeh et al., 2017) with the model trained on NYT as initialization. We use the train set of ROCStories, which contains around 100K short stories (each consists of five sentences) [5].

Preprocessing

We pre-process all documents with Semantic Role Labeling (SRL) (Punyakanok et al., 2004) and Part-of-Speech (POS) tagger (Roth and Zelenko, 1998). We also implement the explicit discourse connective identification module of a shallow discourse parser (Song et al., 2015). Additionally, we utilize within

[4] Available at https://catalog.ldc.upenn.edu/LDC2008T19

[5] Available at http://cs.rochester.edu/nlp/rocstories/

document entity co-reference (Peng et al., 2015a) to produce co-reference chains to get the *new entity* information. To obtain all annotations, we employ the Illinois NLP tools[6].

7.4.2 FES Representation Generation

As shown in Sec. 7.3, each FES representation is built from basic semantic units: frame / entity / sentiment. We describe our implementation details on how we extract these units from text and how we further construct their vector representations respectively.

Frame Abstraction and Enrichment

We directly derive semantic frames from semantic role labeling annotations. As the Illinois SRL package is built upon PropBank frames, we map them to FrameNet frames via VerbNet senses to achieve a higher level of abstraction. The mapping is deterministic and partial[7]. For unmapped PropBank frames, we retain their original PropBank forms. We then enrich the frames by augmenting them to verb phrases. We apply three heuristic rules:

- If a preposition immediately follows a predicate, we append the preposition e.g. *"take over"*.
- If we encounter the role label AM-PRD which indicates a secondary predicate, we append it to the main predicate e.g. *"be happy"*
- If we see the semantic role label AM-NEG which indicates negation, we append "not" e.g. *"not like"*. We further connect compound verbs together as they represent a unified semantic meaning. For this, we apply a rule that if the gap between two predicates is less than two tokens, we treat them as a unified semantic frame defined by the conjunction of the two (augmented) semantic frames, e.g. *"decide to buy"* being represented by "decide.01-buy.01".

To sum up, we employ the same techniques to deal with frames as discussed in Peng and Roth (2016), which allows us to model more fine-grained semantic frames. As an example of this processing step, *"He didn't want to give up."* is represented as "(not)want.01-give.01[up]". Each semantic frame (here, including discourse markers) is represented by a 200-dimensional vector e_f.

Entity Label Assignment

For each entity (here we refer to subject and object of the predicate), we first extract its syntactic head using Collins' Head Rule. To assign entity types, we then check if the head is inside a named entity generated by NER. If so, we directly assign the NER label to this entity. Otherwise, we check if the entity is a pronoun

[6]Available at http://cogcomp.org/page/software/

[7]We use the mapping file http://verbs.colorado.edu/verb-index/fn/vn-fn.xml to do it. For example, "place" and "put" with the same VerbNet sense id "9.1-2" are both mapped to the FrameNet frame "Placing".

	Vocabulary Size				Sequence Size	
	FES	F	E	S	#seq	#token
NYT	4M	15K	100	7	1.2M	25.4M
ROCStories	200K	1K	98	7	100K	630K

Table 7.2: Statistics on FES-LM vocabularies and sequences. We compare FES-LM trained on NYT vs. ROCStories; "FES" stands for unique FES representations while "F" for frame embeddings, "E" for entity representations, and "S" for sentiment representations. "#seq" is the number of sequences, and "#token" is the total number of tokens (FES representations) used for training.

that refers to a person i.e. *I, me, we, you, he, him, she, her, they, them*; in which case, we assign "PER" label to it. For all other cases, we simply assign "ARG" label to indicate the type is unknown.

In order to assign "new entity" labels, we check if the head is inside a mention identified by the co-reference system to start a new co-reference chain. If so, we assign 1; otherwise, we assign 0. On ROCStories dataset, we add an additional rule that all pronouns indicating a person will not be "new entities". This makes the co-reference decisions more robust on short stories.[8]

The entity representation r_e is eventually constructed as a one-hot vector for types of 5 dimensions and an additional dimension for "new entity" information. As we consider both subjects and objects of a frame, r_e is of 12 dimensions in total. If either one of the entities within a frame is missing from SRL annotations, we set its corresponding 6 dimensions as zeros.

Sentiment Representation Generation

We first determine the polarity of a word by a look-up table from two pre-trained sentiment lexicons (Liu et al., 2005; Wilson et al., 2005b). We then count the number of positive words versus negative words to decide the sentiment of a piece of text as detailed in Sec. 7.2. This process is done on text corresponding to each frame, i.e. a sentence or a clause. Since we have two different lexicons, we get two separate one-hot sentiment vectors, each with a dimension of 3. Thus, the sentiment representation is the concatenation of the two vectors, a total dimension of 6.

7.4.3 Neural Language Model Training

For the NYT corpus, we treat each document as a single semantic sequence while on ROCStories, we see each story as a semantic sequence. Additionally, we filter out rare frames which appear less than 20 times in the NYT corpus. Statistics on the eventual FES-LM vocabularies (unique FES representations) and semantic sequences in both datasets are shown in Table 7.2. Note that the number of unique FES representations reflects the richness of the semantic space that we model. On both datasets, it is about 200 times over

[8]The same rule is not applied on news, since pronouns indicating a person can start a co-reference chain in news.

Narrative Cloze Test (Recall@30)	CBOW	SG	LBL
FES-LM	38.9	37.3	**43.2**
FES-LM - Entity	35.3	33.1	38.4
FES-LM - Sentiment	34.9	32.8	36.3

Table 7.3: Quality comparison of neural language models. We report results for narrative cloze test. The evaluation is done on the gold PropBank data (annotated with gold frames). LBL outperforms CBOW and SG. We carry out ablation studies for FES-LM without entity and sentiment aspects respectively.

	CoNLL16 Test			CoNLL16 Blind		
	Explicit	Implicit	Overall	Explicit	Implicit	Overall
Base (Song et al., 2015)*	89.8	35.6	60.4	75.8	31.9	52.3
SemLM (Peng and Roth, 2016)	**91.1**	36.3	61.4	77.3	33.2	53.8
Top (Mihaylov and Frank, 2016)	89.8	**39.2**	**63.3**	78.2	**34.5**	**54.6**
FES-LM (this work)	91.0	37.5	61.8	**78.3**	34.4	54.5
FES-LM - Entity	90.8	37.1	61.6	77.9	34.0	54.1
FES-LM - Sentiment	90.5	36.9	61.3	77.3	33.8	53.9

Table 7.4: Shallow discourse parsing results. With added FES-LM features, we get significant improvement (based on McNemar's Test) over the base system(*) and outperform SemLM, which only models frame information. We also rival the top system (Mihaylov and Frank, 2016) in the CoNLL16 Shared Task (connective sense classification subtask).

what is modeled by only frame representations. At the same time, we do not incur burden on language model training. It is because we do not model unique FES representations directly, and instead we are still operating in the frame embedding space.[9]

We use the OxLM toolkit (Baltescu et al., 2014) with Noise-Constrastive Estimation (Gutmann and Hyvarinen, 2010) to implement the LBL model. We set the context window size to 5 and produce 200-dimension embeddings for FES representations. In addition to learning language model parameters, we also learn frame embeddings e_f along with parameters for W_e (12x200 matrix) and W_s (6x200 matrix).

7.5 Evaluation

We first show that our proposed FES-LM is of high quality in terms of language modeling ability. We then evaluate FES-LM for shallow discourse parsing on news data as well as application for story cloze test on short common sense stories. In all studies, we verify that each semantic aspect contributes to the joint model.

[9]The FES representation space can be seen as entity and sentiment infused frame embedding space.

7.5.1 Quality of FES-LM

To evaluate the modeling ability of different neural language models, we train each variant of FES-LM on NYT corpus and report narrative cloze test results. Here, we choose the Skip-Gram (SG) model (Mikolov et al., 2013c) and Continuous-Bag-of-Words (CBOW) model (Mikolov et al., 2013b) for comparison with the LBL model. We utilize the word2vec package to implement both SG and CBOW. We set the context window size to be 10 for SG and 5 for CBOW.

We employ the same experimental setting as detailed in Peng and Roth (2016). Results are shown in Table 7.3. They confirm that LBL model performs the best with the highest recall for narrative cloze test.[10] Note that the numbers reported are not directly comparable with those in literature (Rudinger et al., 2015; Peng and Roth, 2016), as we model much richer semantics even though the numbers seem inferior. We further carry out ablation studies for FES-LM without entity and sentiment aspects respectively. The results show that sentiment contributes more than entity information.

7.5.2 Application on News

We choose shallow discourse parsing as the task to show FES-LM's applicability on news. In particular, we evaluate on identifying the correct sense of discourse connectives (both explicit and implicit ones). We choose Song et al. (2015), which uses a supervised pipeline approach, as our base system. We follow the same experimental setting as described in Peng and Roth (2016), i.e. we add additional conditional probability features generated from FES-LM into the base system. We evaluate on CoNLL16 (Xue et al., 2016) test and blind sets, following the train and development split from the Shared Task, and report F1 using the official shared task scorer.

Table 7.4 shows the results for shallow discourse parsing with added FES-LM features. We get significant improvement over the base system(*) (based on McNemar's Test) and outperform SemLM, which only utilizes frame information in the semantic sequences. We also rival the top system (Mihaylov and Frank, 2016) in the CoNLL16 Shared Task (connective sense classification subtask). Note that the FES-LM used here is trained on NYT corpus. The ablation study shows that entity aspect contributes less than sentiment aspect in this application.

[10] We also tried Neural-LSTM (Pichotta and Mooney, 2016) and context2vec (Melamud et al., 2016) model, but we cannot get better results.

Baselines		
Seq2Seq		58.0%
DSSM (Mostafazadeh et al., 2016)		58.5%
Seq2Seq with attention		**59.1%**
Individual Aspect	S.	M.V.
F-LM	57.8%	56.3%
E-LM	52.1%	52.6%
S-LM	54.2%	54.9%
Joint Model	S.	M.V.
FES-LM (this work)	**62.3%**	61.6%
FES-LM - Entity	61.5%	61.7%
FES-LM - Sentiment	61.1%	60.9%

Table 7.5: Accuracy results for story cloze text in the unsupervised setting. "S." represents the inference method with the single most informative feature while "M.V." means majority voting. FES-LM outperforms the strongest baseline (Seq2Seq with attention) by 3 points. The difference is statistically significant based on McNemar's Test. Additional ablation studies show that each semantic aspect contributes to the joint model.

7.5.3 Application on Stories

For the story cloze test on the ROCStories dataset. We evaluate in an unsupervised setting, where we disregard the labeled development set and directly test on the test set[11]. We believe this is a better setting to reflect a system's ability to model semantic sequences compared to the supervised setting where we simply treat the task as a binary classification problem with a development set to tune.

We first generate a set of conditional probability features from FES-LM. For each story, we extract semantic aspect information as described in Sec. 7.3 and construct the joint FES representation according to the learned FES-LM. We then utilize the conditional probability of the fifth sentence s_5 given previous context sentences C as features. Suppose the semantic information in the fifth sentence can be represented by $r_{\text{FES\_k}}$, we can then define the features as $p(s_5|C) = p(r_{\text{FES\_k}}|r_{\text{FES\_(k-1)}}, r_{\text{FES\_(k-2)}}, \cdots, r_{\text{FES\_(k-t)}}), \quad t = 1, 2, \cdots, k$. We get multiple features depending on how long we go back in the context in terms of FES representations. Note that one sentence can contain multiple FES representations depending on how many semantic frames it has. For simplicity, we assume a single FES representation $r_{\text{FES\_k}}$ for s_5. In practice, we get at most 12 FES representations as context. We align the features by t, indicating how long we consider the story context. Thus, for each story, we generate at most 12 pairs of conditional probability features. Evey pair of such features can yield a decision on which ending is more probable. Here, we test two different inference methods: a single most informative feature (where we go with the decision made by the pair of features which have the highest ratio) or majority voting based on all feature pairs. Note that we need to re-train

[11]The test set contains $1,871$ four-sentences long stories with two fifth sentence options for each, of which only one is correct; and we report the accuracy.

Ex.5 Correct Prediction due to Frame Information
Story: He didn't know how the television worked. He tried to fix it, anyway. He climbed up on the roof and fiddled with the antenna. His foot slipped on the wet shingles and he went tumbling down.

Correct Prediction: Thankfully, he recovered.
Incorrect Choice: He decided that was fun and to try tumbling again.

Ex.6 Correct Prediction due to Entity Information
Maria smelled the fresh Autumn air and decided to celebrate. She wanted to make candy apples. She picked up the ingredients at a local market and headed home. She cooked the candy and prepared the apples.

Correct Prediction: She enjoyed the candy apples.
Incorrect Choice: Maria's apple pie was delicious.

Ex.7 Correct Prediction due to Sentiment Information
Pam thought her front yard looked boring. So she decided to buy several plants. And she placed them in her front yard. She was proud of her work.

Correct Prediction: Pam was satisfied.
Incorrect Choice: Pam was upset at herself.

Figure 7.4: Examples of stories where FES-LM makes correct predictions for the ending. We use data from ROCStories dataset and predictions come from FES-LM with the inference method of single most informative feature.

FES-LM on the stories (train set of ROCStories, 5-sentence stories, no negative examples provided)[12].

We compare FES-LM with Seq2Seq baselines (Sutskever et al., 2014). We also train the Seq2Seq model on the train set of ROCStories, where we set input as the 4-sentence context and the output as the 5th ending sentence for each story. At test time, we get probability of each option ending from the soft-max layer and choose the higher one as the answer. We use an LSTM encoder (300 hidden units) and decode with an LSTM of the same size. Since it is operated on the word level, we use pre-trained 300-dimensional GloVe embeddings (Pennington et al., 2014) and keep them fixed during training. In addition, we add an attention mechanism (Bahdanau et al., 2014) to make the Seq2Seq baseline stronger. We also report DSSM from Mostafazadeh et al. (2016) as the previously best reported result[13]. To study how each individual aspect affects the performance, we develop neural language models on frames (F-LM), entities (E-LM) and sentiments (S-LM) as additional baseline models separately. We use the same language model training and feature generation techniques as FES-LM. Particularly, for F-LM, it is the same model as FC-SemLM defined in Peng and Roth (2016). Note that individual aspects cannot capture the semantic difference between two given options for all instances. For those instances that the baseline model fails to handle, we set the accuracy as 50% (expectation of random guesses).

[12]It is because of domain difference, e.g. average length of semantic sequence is different (stories are shorter while news are longer, see in Table 7.2).

[13]DSSM's model parameters are trained on the ROCStories corpus while hyper parameters are determined on the development set.

Ex.8 Incorrect Prediction due to Frame Information
Story: My friends all love to go to the club to dance. They think it's a lot of fun and always invite. I finally decided to tag along last Saturday. I danced terribly and broke a friend's toe.

Incorrect Prediction: The next weekend, I was asked to stay home.
Correct Choice: My friends decided to keep inviting me as I am so much fun.

Ex.9 Correct Prediction due to Entity Information
Johnny thought Anita was the girl for him, but he was wrong. He invited her out but she said she didn't feel well. Johnny decided to go to a club, just to drink and listen to music. At midnight, he looked back and saw Anita dancing with another guy.

Incorrect Prediction: Johnny did not ask Anita out again.
Correct Choice: Johnny wanted to ask Anita out again.

Figure 7.5: Examples of stories where FES-LM fails to make correct predictions for the ending. We use data from ROCStories dataset and predictions come from FES-LM with the inference method of single most informative feature.

The accuracy results are shown in Table 7.5. The best result we achieve (62.3%) outperforms the strongest baseline (Seq2Seq with attention, 59.1%). It is statistically significant based on McNemar's Test ($\alpha = 0.01$), illustrating the superior semantic modeling ability of FES-LM. Results are mixed comparing the two inference methods. The ablation study further confirms that each semantic aspect has its worth in the joint model.

7.5.4 Qualitative Analysis

Figure 7.4 shows example stories where FES-LM makes correct predictions for the ending. Ex.5 describes the story of a man hurting himself. According to commonsense knowledge, people usually recover (correct ending) after being hurt and do not repeat their mistake (incorrect ending), which is relied upon frame sequence information. Ex.6 describes the story of Maria making candy apples. The incorrect ending introduces a new entity "apple pie", resulting in topical incoherence. Similarly, Ex.7 describes the story of Pam being proud of her yard work. There is a striking sentimental contrast between the two options ("upset" versus "satisfied"), and FES-LM makes the right prediction based on the consistency of sentiment information.

Figure 7.5 shows examples of stories for which FES-LM could not predict the correct ending. We believe that many of these stories require a deeper understanding of language and commonsense. In Ex. 8, the protagonist accepted an invitation from his friends to go to a club but danced terribly, so he was asked to stay home the next time. To make the correct prediction, the model not only needs to understand that if one does not dance well at a club they are likely to be not invited later, but also that staying home is the same as not getting invited. Similarly, Ex.9 requires identifying that Anita's excuse was a lie indicating her disinterest in Johnny, which makes it unlikely for Johnny to invite her again. It demonstrates that the model needs an deeper level of language and social understanding, i.e. seeing a potential lover with another

person leads to estrangement. Both negative examples reveal the limitation with FES-LM, and it inspires us to further develop a better version of semantic language model (in Chapter 8).

Chapter 8

SemLM with Knowledge

Story understanding entails the need to have an accurate expectation of what events would be described next in text. When humans decide whether a future event is likely to occur or not, it depends on not only the events that have happened earlier, but also *knowledge* gained through human experience. Thus, in this chapter, we gather the knowledge of event causality, i.e. one event leads to another, both statistically and manually. Such knowledge is modeled in an explicit way with frame and entity level abstractions, and is infused into a semantic language model via a joint training and inference procedure. We show that the proposed `KnowSemLM` can better predict future events via applications for story cloze test and referent prediction task compared to models without knowledge; while at the same time `KnowSemLM` still maintains high quality in terms of language modeling ability.

8.1 Motivation

A story can be seen as a description of a series of events. Thus, story understanding requires not only understanding what events have happened in text, but also understanding what events would happen next. For the first part (understanding of events), many past works have been devoted to the task of event extraction (Ji and Grishman, 2008; Huang and Riloff, 2012b; Li et al., 2013; Peng et al., 2016). For the the second part (predicting future events), a natural way is to make predictions based on past events, i.e. utilizing the co-occurence information of events from a large corpus. Relevant contributions have been made in the field of script leaning (Chambers and Jurafsky, 2008; Chambers, 2013; Pichotta and Mooney, 2014, 2016; Peng and Roth, 2016; Peng et al., 2017). However, such co-occurence information is far from being complete for modeling event sequences in natural text (which is generated by humans) for the following reason:

Humans' decision of whether a specific event will occur or not depends on both ***local context*** *(what has happened earlier) and* ***global context*** *(knowledge gained from human experience).*

For example, the following text describes a scenario where someone took a flight.

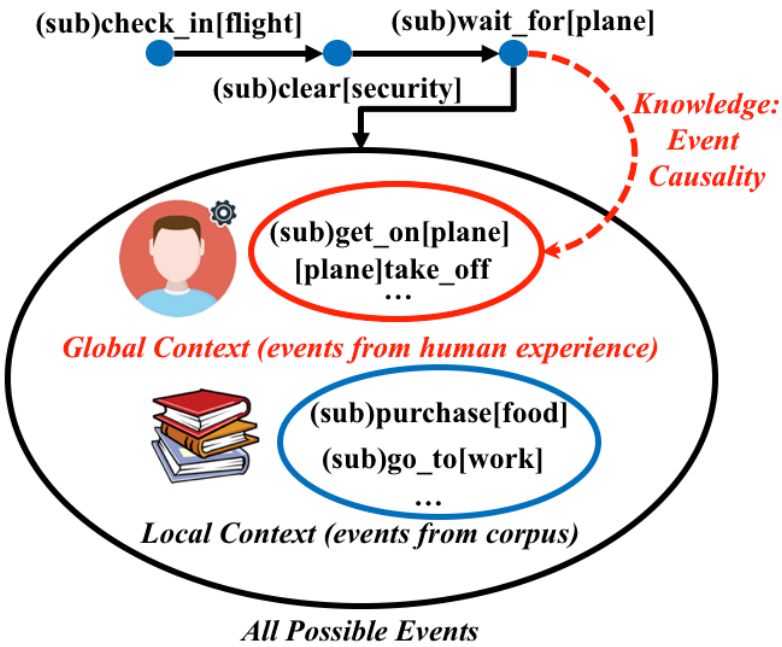

Figure 8.1: Local and Global Context Information when Modeling Event Sequences. Blue dots denote events already described in text. The blue circle indicates local context, i.e. events learned from a large corpus via language models; while the red circle represents global context, i.e. events learned from human experience via knowledge of event causality (which may have overlaps with local context). For event representations, we abstract over the surface forms of semantic frames and entities. The proposed `KnowSemLM` leverages both information to better predict future events.

After my parents wished me goodbye, I checked in at the counter, took my luggage to the security area, got cleared ten minutes in advance, and waited for my plane.

In the above example, there is a series of events, i.e. "check in (a flight)", "clear (security)", "wait for (plane)" and etc. Such an event sequence is semantically sound for any human being who has the experience of traveling by plane. However, it is also an event sequence which does not appear frequently in text.[1] Since language models rely on statistical co-occurrences in the given text to model text/event sequences, they have significant limitations in the ability to encode such common sense knowledge (e.g. "check in (a flight)" event followed by "clear (security)" event). On the contrary, humans learn from everyday activities and are a rich source of such knowledge.

In this chapter, we leverage both the *local* and *global* context information to model event sequences, which leads to more accurate predictions of future events. Both local and global context information is shown in Figure 8.1. The existing event sequence (each event represented by a blue dot) is denoted as "(sub)check_in[flight]", "(sub)clear[security]" and "(sub)wait_for[plane]". Here, "check_in / clear / wait_for" and "[flight] / [plane] / [security]" represent abstractions over the surface forms of semantic frames and entities, respectively. Moreover, they share a common subject denoted by "(sub)". In the above example,

[1]The co-occurence of "check in (a flight)" and "clear (security)" only appears two times in the same document among 20 years of New York Times data. We count with frame and entity level abstraction (see Section 4.3 for details).

the local context is captured by language models for statistical co-occurrences of events, thus generating a distribution over all possible events (in the blue circle), e.g. "(sub)purchase[food]" and "(sub)go_to[work]".

More importantly, the global context comes from the knowledge of event causality (one event leads to another), thus generating a distribution over a (focused) set of expected events learned from human experience (in the red circle). In the above example, one such piece of knowledge can be represented as "(sub)wait_for[plane] ⇒ (sub)get_on[plane]", which means that one has to wait for a plane before getting on it (red dashed arrow in Figure 8.1). Note that such causality links have directions, and one event might lead to multiple possible resulting events, e.g. one has to wait for plane before the plane can take off "(sub)wait_for[plane] ⇒ [plane]take_off". Meanwhile, multiple events can lead to the same event, e.g. before getting on a plane, one needs to clear security "(sub)clear[security] ⇒ (sub)get_on[plane]".

Thus, we propose `KnowSemLM`, a knowledge infused semantic language model. It combines knowledge from external sources (in the form of event causality) with the basic semantic language model trained from the given text corpus (Peng et al., 2017). In this model, we assume that each event is either generated based on a piece of knowledge or generated from the semantic language model. When predicting future events, i.e. the inference procedure, at each time step, we generate a distribution over events from knowledge as well as a distribution over events from the semantic language model. We also learn a binary variable to choose between events from either one of the distributions. In such a way, the proposed `KnowSemLM` has the ability to generate events sequences based on both local and global context, and better imitate the story generation process of a human being. This knowledge infused semantic language operates on abstractions over the surface forms of semantic frames and entities. We associate each semantic unit (abstracted frames, entities, and etc.) with an embedding and construct a joint embedding space for each event. We train `KnowSemLM` on a large corpus with the same embedding setting for events involved in knowledge. Additionally, we mine the event causality knowledge both statistically from the training corpus and manually for constrained test domains.

We evaluate `KnowSemLM` on applications for referent prediction and story prediction tasks. In both cases, we model text as event sequences, and apply trained `KnowSemLM` to calculate conditional probabilities of future events given text and knowledge. We show that those conditional probabilities can either be directly used for task inference, or be plugged into a base model as additional features; thus outperforming competitive results from models without the use of knowledge. In addition, we demonstrate the language modeling ability of `KnowSemLM` through both quantitative and qualitative analysis.

The main contributions of our work can be summarized as follows:

1. We formally define the knowledge (global context) used in story generation as event causality.

2. We propose `KnowSemLM` to integrate such event causality knowledge into semantic language models.

3. We demonstrate the effectiveness of the proposed `KnowSemLM` when predicting future events via benchmark tests and analysis.

8.2 Event and Knowledge Modeling

In order to model event sequences in text, we first introduce the event representation that is being used in this chapter, which is constructed from abstractions of basic semantic units, including semantic frames, entities and sentiments. We then define how we model event causality knowledge between events, which is represented in an explicit fashion while re-using event representations.

8.2.1 Event Representation

To preserve the full semantic meaning of events, we need to consider multiple semantic aspects: semantic frames, entities, and sentiments. We adopt the event representation proposed in Peng et al. (2017), which is built upon abstractions of three basic semantic units.

Semantic frame

Semantic frame is originally defined in Fillmore (1976), which can be simplified as a composition of a predicate and its corresponding argument participants. In this chapter, we require the predicate to be disambiguated to a specific sense, thus each frame can be uniquely represented by its predicate sense. These frames provide a good level of generalization as each frame can be instantiated into various surface forms in natural texts. The arguments of the frames are modeled as entities, as described next. Additionally, we extend the frame definitions to include explicit discourse markers since they model relationships between frames (Peng and Roth, 2016). We use surface forms to represent discourse markers because there is only a limited set. We assign each of these frames (and discourse markers) with an unique embedding r_f of the same dimension, which is to be learned during language model training.

Entity

We consider the subject and object of a predicate as the essential entity information for modeling events. To achieve a higher level of abstraction, we model entity types instead of entity surface forms. We choose to assign entities with labels produced by Named Entity Recognition (NER). Besides type information, we also embed the most basic entity co-reference information, i.e anaphrocity - whether or not an entity has co-referred antecedents in text. Thus, we can encode this binary information as an additional dimension in the entity representation. Thus, we formally define r_e as the entity representation. It is the concatenation

of two entity vectors r_{sub} and r_{obj} for subject entity and object entity respectively. Both r_{sub} and r_{obj} are constructed as a one hot vector to represent an entity type, plus an additional dimension indicating whether or not it is a new entity (1 if it is new).

Sentiments

For a piece of text, we can assign a sentiment value to it. It can either be positive, negative, or neutral. In order to decide which one is the most appropriate, we first use a look-up table from word lexicons to sentiment, and then count the number of words which corresponds to positive (n_{pos}) and negative (n_{neg}) sentiment respectively. If $n_{pos} > n_{neg}$, we determine the text as positive; and if $n_{pos} < n_{neg}$, we assign the negative label; and if the two numbers equal, we deem the text as neutral. We use one hot vector for three sentiment choices, and define sentiment representation as r_s.

Joint Representation

In a nutshell, the event representation is a combination of the above three semantic elements.

Steven Avery committed murder. He was arrested, charged and tried.

For example, the event representations of the above text would be

PER[new]-commit.01-ARG[new](NEG), ARG[new]-arrest.01-PER[old](NEU),
ARG[new]-charge.05-PER[old](NEU), ARG[new]-try.01-PER[old](NEG).

Here, "commit.01", "arrest.01" and so on represent disambiguated predicates. The arguments (subject and object) of a predicate are denoted with NER types ("PER, LOC, ORG, MISC") or "ARG" if unknown, along with a "[new/old]" label indicating if it is first mentioned in the sequence. Additionally, the sentiment of a frame is represented as positive (POS), neural (NEU) or negative (NEG).

We formally define such an explicit and abstracted event as e. Computationally, the vector representation of an event e^{vec} is built in a joint semantic space.

$$e^{\text{vec}} = r_f + W_e r_e + W_s r_s$$

During language model training, we learn frame embeddings r_f as well as the transforming matrices W_e and W_s.

8.2.2 Knowledge: Causality between Events

In this chapter, we model the knowledge gained from human experience as pre-determined relationship between events. Since we are modeling event sequences, the knowledge of one event leads to another is very

important, hence event causality. Here we formally define a piece of event knowledge as

$$e_x \Rightarrow e_y,$$

meaning that the *outcome* event e_y is a possible result of the *casual* event e_x. Note that event causality here is directional, and one event may lead to multiple different outcomes. We group all event knowledge pairs with the same casual event, thus event e_x can lead to a set of events

$$e_x \Rightarrow \{e_{y_1}, e_{y_2}, e_{y_3}, \cdots, e_{y_m}\}.$$

It can be seen as a tree structure, where the casual event is the root and all outcome events are leaf nodes. To facilitate the training and inference over such knowledge, we store all such event causality tree structures in a knowledge base KB_{EC} (different trees may have different number of leaf nodes, indicated by indices - $m_1, m_2, m_3, \cdots$):

$$\text{KB}_{\text{EC}} = \begin{pmatrix} e_{x_1} \Rightarrow \{e_{y_1}, e_{y_2}, e_{y_3}, \cdots, e_{y_{m_1}}\}, \\ e_{x_2} \Rightarrow \{e_{y'_1}, e_{y'_2}, e_{y'_3}, \cdots, e_{y'_{m_2}}\}, \\ e_{x_2} \Rightarrow \{e_{y''_1}, e_{y''_2}, e_{y''_3}, \cdots, e_{y''_{m_3}}\}, \\ \cdots \end{pmatrix}.$$

To build such a knowledge base, we have two different ways to get such event causality pairs: either we generate them manually based on our common sense knowledge, or we can automatically detect them from text. For both cases, we still represent events with abstractions as detailed in Sec. 2.1. Note that for knowledge representation, we utilize events in an explicit way (instead of their vector representations). Thus, during the training and inference process for the proposed `KnowSemLM`, we match for the exact *casual* event to activate the use of knowledge; while the event abstractions provide a level of generalization. For example, the knowledge of "an attacker is likely to be arrested" can be formally denoted as "PER[*]-attack.01-*[*](*) ⇒ *[*]-arrest.01-PER[old](*)". Here, "*" indicates that it can match for any semantic element in that place (entity type, anaphoricity or sentiment), which allows for a higher level of abstraction. More details how we inject such knowledge and gather them can be referred to Sec. 3.2 and 4.3, respectively. [2]

[2]The event causality knowledge examples in Sec. 1 are generated manually from the InScript Corpus (Modi et al., 2017). Since the data provides event templates and corresponding event annotations, we utilize abstractions over such event templates as event representations.

8.3 Knowledge Infused SemLM

In this section, we first re-visit the base semantic language model, i.e. FES-RNNLM proposed in Peng et al. (2017). To facilitate simpler training and inference process, we utilize the *Recurrent Neural Network Language Model* (RNNLM) (Bengio et al., 2003) instead of the *Log-Bilinear Language model* (LBL) (Mnih and Hinton, 2007). We then describe the procedures of how we inject event causality knowledge into FES-RNNLM to obtain the proposed `KnowSemLM`.

8.3.1 FES-RNNLM

To model semantic sequences and train the joint event representations in Sec. 2.1, we build neural language models over such sequences. Theoretically, we can utilize any existing neural language model. However, since we require the use of event causality knowledge to be based on past events, we need to implement a language model where the generation of future events is only dependent on past events. Thus, we choose to implement RNNLM in this chapter.

For ease of explanation, we assume that a semantic sequence of joint event representations is $[e_1, e_2, e_3, \cdots, e_k]$, with e_t being the t_{th} event in the sequence. Thus, we model the conditional probability of an event e_t given its context $[e_1, e_2, \cdots, e_{t-1}]$:

$$\begin{aligned} p_{\text{lm}}(e_t|e_1, \cdots, e_{t-1}) &= \text{softmax}(W_s h_t + b_s) \\ &= \frac{\exp(e_t^{\text{vec}}(W_s h_t + b_s))}{\sum_{e \in \mathcal{V}} \exp(e^{\text{vec}}(W_s h_t + b_s))}. \end{aligned}$$

Note that the softmax operation is carried out over the event vocabulary $\mathcal{V}$, i.e. all possible events in the language model. Moreover, the hidden layer h_t in RNN is computed as:

$$h_t = \phi(e_t^{\text{vec}} W_i + h_{t-1} W_h + b_h).$$

Here, ϕ is the activation function. For language model training, we learn parameters W_s, b_s, W_i, W_h, b_h, and maximize the overall sequence probability:

$$\prod_{t=1}^{k} p_{\text{lm}}(e_t|e_1, e_2, \cdots, e_{t-1}).$$

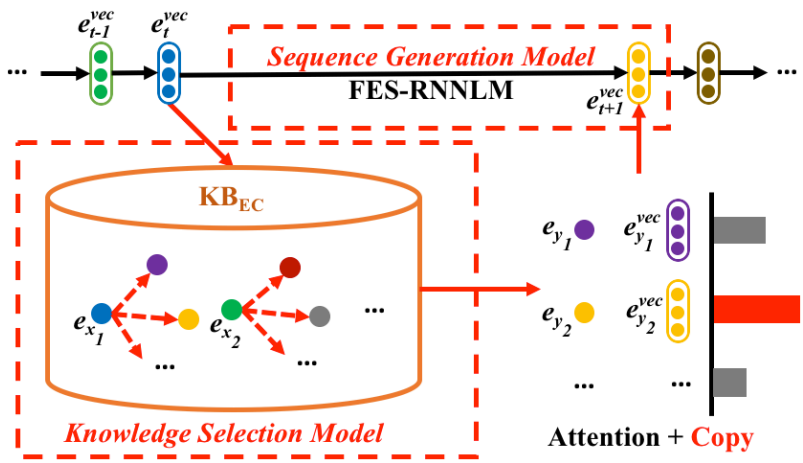

Figure 8.2: Overview of the Computational Workflow for the proposed `KnowSemLM`. There are two key components: 1) a knowledge selection model, which activates the use of knowledge based on matching casual event and produce a distribution over outcome events via attention; 2) a sequence generation model, which takes input from both the knowledge selection model and the base semantic language model (FES-RNNLM) to generate future events via a copying mechanism. Note that the single dots indicate explicit event representations while three consecutive dots stand for event vectors.

8.3.2 `KnowSemLM`

In Figure 8.2, we show the computational workflow of the proposed `KnowSemLM`. There are two key components: 1) a knowledge selection model, which activates the use of knowledge based on matching casual event and produce a distribution over outcome events; 2) a sequence generation model, which takes input from both the knowledge selection model and the base semantic language model (FES-RNNLM) to generate future events via a copying mechanism. Now, we describe the details of the two components.[3]

Knowledge Selection Model

For an event in the sequence e_t, we first match it with all possible casual events $\{e_x\}$ in the event causality knowledge base KB_{EC}. Thus, from the knowledge base, we get a list of outcome events $\mathcal{V}_y \triangleq \{e_{y_1}, e_{y_2}, \cdots\}$. Note that the matching process here is using the explicit event representations from Sec. 2.1 instead of their corresponding vector representations. Additionally, the event level semantic abstractions over frames, entities and sentiments allow we activate the use of event causality knowledge in a general way. Moreover, we designate a "dummy" outcome event for cases where no casual event can be matched.

Since the use of event causality knowledge (i.e. how likely an outcome event shall happen) is related to what has been described in text, we model the conditional probability of e_y from knowledge base given the

[3]The proposed computational framework of `KnowSemLM` is similar to DynoNet proposed in He et al. (2017). Compared to Dynonet, the knowledge base utilized here is simpler, and `KnowSemLM` operates on event level representations rather than on tokens.

context of $e_1, e_2, \cdots, e_t$, thus

$$\begin{aligned} p_{\text{kn}}(e_y|e_1, e_2, \cdots, e_t) &= \text{softmax}(W_a h_t) \\ &= \frac{\exp(e_y^{\text{vec}} W_a h_t)}{\sum_{e \in \mathcal{V}_y} \exp(e^{\text{vec}} W_a h_t)}. \end{aligned}$$

Here, we use the attention mechanism (Bahdanau et al., 2014) via learned attention parameter W_a; and apply it on the hidden layer h_t, which embeds information from all previous events in the sequence. As the softmax here is operated over the set of outcome events $\mathcal{V}_y$, we produce a distribution over the possible outcome events.

Sequence Generation Model

The base semantic language model produces a distribution over events from the language model vocabulary, which represents *local* context; while the knowledge selection model generates a set of outcomes events with a probability distribution, which represents *global* context of event causality knowledge. Thus, the sequence generation model combines the local and global context for generating future events. So, we model the conditional probability of event e_{t+1} given context:

$$p(e_{t+1}|\text{Context}) = p(e_{t+1}|e_1, e_2, \cdots, e_t, \text{KB}_{\text{EC}}).$$

This overall distribution is computed via a copying mechanism (Jia and Liang, 2016):

$$\begin{cases} p(e_{t+1} = e_i \in \mathcal{V}|\text{Context}) &= (1-\lambda) p_{\text{lm}}(e_i) \\ p(e_{t+1} = e_y \in \mathcal{V}_y|\text{Context}) &= \lambda p_{\text{kn}}(e_y). \end{cases}$$

Here, λ is a learned scaling parameter to choose between events from language model vocabulary and events from event causality knowledge.

8.4 Building `KnowSemLM`

In this section, we show how we build `KnowSemLM` from scratch by laying out the details for 1) data source and preprocessing steps 2) event-level abstractions 3) knowledge mining and 4) language model training.

8.4.1 Dataset and Preprocessing

Dataset We use the New York Times (NYT) Corpus[4] (from year 1987 to 2007) as the training corpus. It contains over 1.8M documents in total.

Preprocessing We pre-process all training documents with Semantic Role Labeling (SRL) (Punyakanok et al., 2004) and Part-of-Speech (POS) tagger (Roth and Zelenko, 1998). We also implement the explicit discourse connective identification module of a shallow discourse parser (Song et al., 2015). Additionally, we utilize within document entity co-reference (Peng et al., 2015a) to produce co-reference chains to get the anaphrocity information. To obtain all annotations, we employ the Illinois NLP tools[5].

8.4.2 Event Abstractions

Similar to the procedures implemented in Peng et al. (2017), we obtain event representations from text with frame, entity and sentiment level abstractions. Here's a brief summary.

Frame Abstraction and Enrichment

We directly derive semantic frames from SRL annotations. We map PropBank frames to FrameNet frames via VerbNet senses to achieve a higher level of abstraction. We then enrich the frames by augmenting them to verb phrases by applying heuristic rules: 1) append the preposition that follows a predicate, e.g. *"take over"*; 2) append the secondary predicate to the main predicate, e.g. *"be happy"*; 3) append "not" if there is negation, e.g. *"not want"*. We further connect compound verbs together (if the gap between two predicates is less than two tokens) as they represent a unified semantic meaning, e.g. *"decide to buy"* being represented by "decide.01-buy.01". As an example of this processing step, *"He didn't want to give up."* is represented as "(not)want.01-give.01[up]".

Entity Label Assignment

For each entity (subject/object of a predicate), we first get its syntactic head using Collins' Head Rule. We then check if the head is inside a named entity generated by NER: if so, we directly assign the NER label; otherwise, we check if the entity is a pronoun that refers to a person, where we assign "PER" label to it. For all other cases, we simply assign "ARG" label to indicate unknown entity type. For anaphoricity information, we check if the head is inside a mention identified by the co-reference system to start a new co-reference chain. If so, we assign 1; otherwise, we assign 0. If either one of the entities within a frame is missing from SRL annotations, we set its corresponding vector as a zero vector.

Sentiment Representation Generation

[4] Available at https://catalog.ldc.upenn.edu/LDC2008T19

[5] Available at http://cogcomp.org/page/software/

We determine the polarity of a word by a look-up table from two pre-trained sentiment lexicons (Liu et al., 2005; Wilson et al., 2005b). We then count the number of positive words versus negative words to decide the sentiment of a piece of text as detailed in Sec. 2.1. This process is done on text corresponding to each frame, i.e. a sentence or a clause.

8.4.3 Knowledge Mining

We can generate the event causality knowledge defined in Sec. 2.2 either statistically from a large corpus or manually from a given scenario.

Statistical Way

Part of the human knowledge can be mined from text itself. Based on this intuition, we attempt to find event causality pairs mentioned in text. Since discourse connectives are important for relating different text spans, here we carefully select discourse connectives which can indicate a "cause-effect" situation. For example, "The police arrested Jack *because* he killed someone." In this sentence, the discourse connective (*because*) evokes the "Cause" discourse relation. This allows readers to relate its two associated events, "The police arrested Jack" and "he killed someone", gaining the knowledge of "the person who kills can be arrested", which can be represented as "PER[*]-kill.01-*[*](*) ⇒ *[*]-arrest.01-PER[old](*)" according to the abstractions and definitions specified in Sec. 2.

In practice, we decide on 22 "cause-effect" connectives/phrases (such as "because", "due to", "in order to", etc.). We then extract all event pairs connected by such connectives from the NYT training data, and abstract over the their surface forms to get the event level representations. Finally, We filter cases where the direction of the event causality pairs is unclear from a statistical standpoint. Specifically, we calculate the ratio of counts of one direction over another, i.e.

$$\theta = \frac{\#(e_x \Rightarrow e_y)}{\#(e_y \Rightarrow e_x)}.$$

If $\theta > 2$, then we store $e_x \Rightarrow e_y$ as knowledge, while removing the case of $e_y \Rightarrow e_x$; If $\theta < 0.5$, then the direction of the knowledge is reversed, and we only keep $e_y \Rightarrow e_x$. In the case of $0.5 < \theta < 2$, we filter both event causality pairs since we are unsure of the knowledge statistically.

After filtering, we automatically get 8,293 different pairs of event causality pairs (without human efforts). According to Sec.2, we merge them with the common casual event, thus getting a total of 2,037 casual events (trees); and on average, each casual event has 4 possible outcome events.

Manual Way

Besides mining knowledge automatically from text corpus, we can also take full advantage of human input in some practical situations. Here, for the InScript Corpus (Modi et al., 2017), it specifies 10 every day scenarios, e.g. "Bath", "Flight", "Haircut", "Grocery", etc. In each scenario, it also provides event templates and the corresponding event template annotations for the text. Thus, we can directly apply our human knowledge based on such templates (which can be naturally converted to the event representation defined in Sec. 2. Examples of such generated event causality knowledge can be referred back to Sec. 1, e.g. "(sub)wait_for[plane] ⇒ (sub)get_on[plane]". In total, we manually generate 875 event causality pairs and group them with 121 casual events. So, on average each casual event can lead to 7 different outcome events.

8.4.4 Model Training

We train `KnowSemLM` on the NYT training corpus along with the event causality knowledge base $\mathrm{KB}_{\mathrm{EC}}$, which is built from knowledge automatically mined from the NYT corpus itself. Based on the formulation in Sec. 3, we apply the overall sequence probability as the training objective:

$$\prod_{t=1}^{k} p_{\texttt{KnowSemLM}}(e_t|e_1, e_2, \cdots, e_{t-1}, \mathrm{KB}_{\mathrm{EC}}).$$

For the sequence generation model, we use implement the Long Short-Term Memory (LSTM) network with a layer of 64 hidden units while the dimension of the input event vector representation is 200. Because we carry out the same event level abstractions as in Peng et al. (2017), the event vocabulary is the same with 4M different events[6].

8.5 Experiments

We first show that the proposed `KnowSemLM` can achieve better performance for story cloze test and referent prediction tasks compared to models without the use of knowledge. We also evaluate the language modeling ability of `KnowSemLM` through quantitative an qualitative analysis.

8.5.1 Application for Story Prediction

Task Description and Setting

For story cloze test, we use the benchmark ROCStories dataset (Mostafazadeh et al., 2017), and follow the test setting in Peng et al. (2017). For each test instance, we are given a four-sentence and the system needs to

[6]More details can be referred to Table 2 in Peng et al. (2017)

Baselines	Accuracy	
Seq2Seq		58.0%
DSSM (Mostafazadeh et al., 2016)		58.5%
Seq2Seq with attention		59.1%
Base Model w/o Knowledge	S.	M.V.
FES-LM (Peng et al., 2017)	62.3%	61.6%
Knowledge Model	S.	M.V.
KnowSemLM	**66.5%**	63.1%

Table 8.1: Accuracy results for story cloze test in the unsupervised setting. "S." represents the inference method with the single most informative feature while "M.V." means majority voting. `KnowSemLM` outperforms both the baselines and the base model without the use of knowledge.

predict the correct following fifth sentence from two choices. The ROCStories dataset provides around 100k five-sentence stories as domain training data, and also a development set with the negative choices of ending sentences. Instead of treating the task as a supervised binary classification problem with a development set to tune, we evaluate `KnowSemLM` in an unsupervised fashion where we disregard the labeled development set and directly test on the test set. In such a way, we can directly compare with the FES-LM model proposed in Peng et al. (2017), which is base model of `KnowSemLM` without the use of knowledge. Thus, similar to the training of FES-LM, we also fine tune `KnowSemLM` on the in-domain short story training data, with the model trained on NYT corpus as initialization.

Application of `KnowSemLM`

For each test story, We generate a set of conditional probability features from `KnowSemLM`. We first construct the event level representation as described in Sec. 4.2. We then utilize the conditional probability of the fifth sentence given previous context sentences and the knowledge base KB_{EC} as features. Note that the event causality knowledge we used here for both training and testing is generated automatically from NYT corpus specified in Sec. 4.3 (the Statistical Way). We get multiple features depending on how long we go back in the context in terms of events. In practice, we get at most 12 events as context since one sentence can contain multiple events depending on how many semantic frames it has. Thus, for each story, we generate at most 12 pairs of conditional probability features from two given choices. Evey pair of such features can yield a decision on which ending is more probable. Here, we test two different inference methods: a single most informative feature (where we go with the decision made by the pair of features which have the highest ratio) or majority voting based on the decision made jointly by all feature pairs.

Results

The accuracy results are shown in Table 8.1. We compare `KnowSemLM` with Seq2Seq baselines (Sutskever et al., 2014) and Seq2Seq with attention mechanism (Bahdanau et al., 2014) following the setting in Peng et al. (2017). We also report DSSM from Mostafazadeh et al. (2016) as the original reported result. `KnowSemLM`

Method	Accuracy
Base (Modi et al., 2017)	62.65%
EntityNLM (Ji et al., 2017)	74.23%
Re-Base	60.58%
Re-Base w/ FES-RNNLM	63.79%
Re-Base w/ `KnowSemLM`	**75.18**%

Table 8.2: Accuracy results for the referent prediction task on InScript Corpus. We re-implemented the base model (Modi et al., 2017) as "Re-base", and apply `KnowSemLM` to add additional features. "Re-Base w/ FES-RNNLM" is the ablation study where no event causality knowledge is used. Even though "Re-base" model performs not as good as the original base model, we achieve the best performance with added `KnowSemLM` features.

outperforms both the baselines and the base model without the use of knowledge, i.e. FES-LM. The best performance achieved by `KnowSemLM` uses single most informative feature (the conditional probability depending on only the nearest preceding event and event causality knowledge).

8.5.2 Application for Referent Prediction

Task Description and Setting

For referent prediction task, we follow the setting in Modi et al. (2017), where the system predicts the referent of an entity (or a new entity) given the preceding text. Note that this task makes a forward prediction based on only the left context, which is different from coreference resolution, where the system is provided with both left and right contexts of a mention. Additionally, compared to language modeling, this task only requires predicting entities. The task is evaluated on the InScript Corpus, which contains a group of documents where events are manually annotated according to pre-defined event templates. Each document contains one entity (its surface form is masked as XXX) which needs to be resolved. The InScript Corpus can be divided into 10 situations and is split into standard training, development, and testing sets. We re-train `KnowSemLM` on the InScript Corpus training set.

Application of `KnowSemLM`

For each test case (i.e. an entity inside a document), each candidate choice will be represented as a different event representation. Note that the event representation here comes from the event templates defined in the InScript Corpus. In the meantime, we can extract the event sequence from the preceding context. Thus, we can apply `KnowSemLM` to compute the conditional probability of the candidate event e_{t+1} given the event sequence and the event causality knowledge:

$$p_k(e_{t+1}|e_{t-k}, e_{t-k+1}, \cdots, e_t, \mathrm{KB}_{\mathrm{EC}}).$$

Perplexity	
FES-RNNLM	121.8
`KnowSemLM`	120.7
Narrative Cloze Test (Recall@30)	
FES-RNNLM	47.9
`KnowSemLM`	49.3

Table 8.3: Results for Perplexity and Narrative Cloze Test. Both studies are conducted on the NYT hold-out data. "FES-RNNLM" represents as the semantic language model without the use of knowledge. The numbers shows that `KnowSemLM` has lower perplexity and higher recall on narrative cloze test; which demonstrates the contribution of the infused event causality knowledge.

Here, knowledge in KB_{EC} is generated manually from event templates specified in Sec. 4.3. Moreover, index k decides how far back we consider the preceding event sequence, and $k \in \{1, 2, \cdots, t-1\}$. We then add this set of conditional probabilities as additional features in a base model to train a classifier to predict the right referent. We re-implemented the linear model proposed in Modi et al. (2017), namely "Re-base".

Results

The accuracy results are shown in Table 8.2. We compare with the original base model as well as the EntityNLM proposed in Ji et al. (2017) as baselines. Our re-implemented base model ("Re-base") does not perform as good as the original model. However, with the help of additional features from FES-RNNLM, we outperform the base model. More importantly, with additional features from `KnowSemLM`, we achieve the best performance and beat the EntityNLM system. This demonstrates the importance of the manually added event causality knowledge, and the ability of `KnowSemLM` to successfully capture it.

8.5.3 Analysis of `KnowSemLM`

We conduct both quantitative and qualitative analysis of `KnowSemLM`. First, to evaluate the language modeling ability of `KnowSemLM`, we report perplexity and narrative cloze test results. We employ the same experimental setting as detailed in Peng and Roth (2016) on the NYT hold-out data. Results are shown in Table 8.3. Here, "FES-RNNLM" serves as the semantic language model without the use of knowledge for the ablation study. The numbers shows that `KnowSemLM` has lower perplexity and higher recall on narrative cloze test; which demonstrates the contribution of the infused event causality knowledge.

We also gather the statistics to analyze the usage of event causality knowledge in `KnowSemLM`. We compute two key values: 1) average number of times a casual event match is found in the event causality knowledge base per event (so that we can potentially use the outcome events to predict), i.e. "Match/Event"; 2) average number of times we actually generate event predictions from the outcome events of the knowledge base (result of the final probability distribution), i.e. "Activation/Event". We get the statistics on both NYT

	Match/Event	Activation/Event	λ
NYT	0.13	0.03	0.36
InScript	0.82	0.28	0.46

Table 8.4: Statistics for the use of event causality knowledge in `KnowSemLM`. We gather the statistics for both NYT and InScript Corpus. "Match/Event" represents average number of times a casual event match is found in the event causality knowledge base per event; while "Activation/Event" stands for the average number of times we actually generate event predictions from the outcome events of the knowledge base. In addition, we believe the ratio of "Activation/Event" over "Match/Event" co-relates with the scaling parameter λ.

and InScript Corpus, and associate the numbers with the scaling parameter λ in Table 8.4. The frequency of event matches and event activations from knowledge are both much lower in NYT than in InScript. Moreover, we can compute the chance of an outcome event being used as the prediction when it participates in the probability distribution. On NYT, it is $0.03/0.13 = 23\%$; while on InScript, it is $0.28/0.82 = 34\%$. We believe such chance co-relates with the scaling parameter λ. Note that the explicit matching of the casual events limits the frequency of usage for event causality knowledge. Theoretically, we can make relax the matching condition according to similarities between event vector representations. However, in practice, we do not find that it can further improve the performance of the downstream applications (e.g. story prediction and referent prediction).

For qualitative analysis, we provide a comparative example between `KnowSemLM` and FES-RNNLN in practice. The system is fed into the following input:

... Jane wanted to buy a new car. She had to borrow some money from her father. ...

So, on an event level, we abstract the text as "PER[new]-want.01-buy.01-ARG[new](NEU), PER[old]-have.04-borrow.01-ARG[new](NEU)". For FES-RNNLM, the system predicts the next event as "PER[old]-sell.01-ARG[new](NEU)" since in training date, there are many co-occurrences between the "borrow" event and "sell" event (coming from financial news articles in NYT). In contrast, for `KnowSemLM`, since we have the knowledge "PER[*]-borrow.01-ARG[*](*) $\Rightarrow$ PER[old]-return.01-ARG[old](*)", meaning that something borrowed by someone is likely to be returned, the predicted event would be "PER[old]-return.01-ARG[old](NEU)". This is closer to the real following text semantically: *She promised to return the money once she got a job.*. In a nutshell, `KnowSemLM` works in situations where they require knowledge given that 1) the required knowledge is stored in the event causality knowledge base, and 2) the training data contains scenarios where required knowledge is put into use.

Chapter 9

Conclusion

9.1 Summary

In this dissertation, towards the goal of building an AI system that can understand stories from text input, we make contributions on three fronts:

1. We investigate the optimal way to model events in text via an intermediate structured representation from semantic role labeling.

2. We propose that we can model a sequence of events as a semantic language model with the balance of generality and specificity.

3. We improve event sequence modeling by joint modeling of semantic information and incorporating explicit background knowledge.

In Chapter 3, we tackle the event extraction problem, where we recognize and categorize events, by developing an event detection system with minimal supervision, in the form of a few event examples. Instead of supervised approaches which tend to over-fit on small data, we transform the detection problem into a semantic similarity problem between event mentions and event types.

In Chapter 4, we study an important linguistic phenomena for event arguments, i.e. entity co-reference. We improve on previous approaches via better entity co-reference decisions using a joint model that accounts for co-reference, mention heads and background knowledge.

In Chapter 5, we start to model event sequences by looking into the simplest form - event co-reference. Following the similar idea of dataless classification, we make better event co-reference decisions based on measuring semantic similarities between event mentions in an unsupervised fashion.

In Chapter 6, we develop two basic semantic language models that capture semantic frame chains and discourse information while abstracting over the specific mentions of predicates and entities.

Since the interactions among different semantic aspects also contribute to modeling a story's semantics, in Chapter 7, we further jointly model frames, entities and sentiments along with discourse information,

yielding joint representations of all these semantic aspects.

Finally, in Chapter 8, we enhance our modeling of event sequences by incorporating explicit knowledge (beyond the given text). Humans' understanding of whether a specific event will occur or not depends not only on what has happened earlier in the event sequence, but also on background knowledge. Built upon this inspiration, we inject such knowledge into our semantic language modeling.

The quality of the proposed semantic language models is evaluated both intrinsically, using perplexity and narrative cloze tests and extrinsically – downstream applications such as co-reference, discourse parsing, and story cloze test.

9.2 Future Directions

The thesis shows several promising future research directions. We outline them here.

1. Open domain event extraction: given any set of event types, accurately extract events from text along with the corresponding event arguments, e.g. agents, spatial/temporal argument. Our work in this thesis shows that the necessary supervision needed here is a few event examples for each type. However, un-answered questions include: 1) How to ensure a satisfying performance? 2) How to ensure the variety of event examples so that we do not have duplicates from a semantic representation perspective? 3) How to improve the intrinsic event representation? 4) Is there a better way to calculate similarities between event representations.

2. Story generation: given an outline of a story, generate natural text to tell the story. Here, the "story outline" can be defined as event sequences where a few of the event arguments are specified. How can we instantiate each event (including instantiating event arguments)? What other semantic aspects we need to consider? Is there a better language modeling tool that can be utilized?

References

Bagga, A. and Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. In *ACL-COLING*.

Bagga, A. and Baldwin, B. (1999). Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *ACL-COLING*.

Balasubramanian, N., Soderland, S., Etzioni, O., et al. (2012). Rel-grams: a probabilistic model of relations in text. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Association for Computational Linguistics.

Balasubramanian, N., Soderland, S., Mausam, and Etzioni, O. (2013). Generating coherent event schemas at scale. In *EMNLP*.

Baltescu, P., Blunsom, P., and Hoang, H. (2014). Oxlm: A neural language modelling framework for machine translation. *The Prague Bulletin of Mathematical Linguistics*.

Bamman, D. and Smith, N. A. (2014). Unsupervised discovery of biographical structure from text. *TACL*, 2:363–376.

Bansal, M. and Klein, D. (2012). Coreference semantics from web features. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Bejan, C. A. (2008). Unsupervised discovery of event scenarios from texts. In *FLAIRS Conference*, pages 124–129.

Bejan, C. A. and Harabagiu, S. (2010). Unsupervised event coreference resolution with rich linguistic features. In *ACL*.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *JMLR*.

Bengtson, E. and Roth, D. (2008). Understanding the value of features for coreference resolution. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Biran, O. and McKeown, K. (2013). Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 69–73.

Björkelund, A. and Kuhn, J. (2014). Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*.

Braud, C. and Denis, P. (2015). Comparing word representations for implicit discourse relation classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2201–2211.

Braud, C. and Denis, P. (2016). Learning connective-based word representations for implicit discourse relation identification. In *Empirical Methods on Natural Language Processing*.

Bronstein, O., Dagan, I., Li, Q., Ji, H., and Frank, A. (2015). Seed-based event trigger labeling: How far can event descriptions get us? In *ACL*.

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., and Roossin, P. (1990). A statistical approach to language translation. *Computational Linguistics*.

Brown, P., Pietra, V. D., deSouza, P., Lai, J., and Mercer, R. (1992). Class-based n-gram models of natural language. *Computational Linguistics*.

Cai, Z., Tu, L., and Gimpel, K. (2017). Pay attention to the ending: Strong neural baselines for the roc story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 616–622.

Cardie, C. and Pierce, D. (1998). Error-driven pruning of Treebanks grammars for base noun phrase identification. In *Proceedings of ACL-98*.

Chambers, N. (2013). Event Schema Induction with a Probabilistic Entity-Driven Model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 1797–1807.

Chambers, N. and Jurafsky, D. (2008). Unsupervised learning of narrative event chains. In *ACL*.

Chambers, N. and Jurafsky, D. (2009a). Unsupervised learning of narrative schemas and their participants. In *ACL*, volume 2, pages 602–610.

Chambers, N. and Jurafsky, D. (2009b). Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, ACL 2009*, pages 602–610.

Chang, C.-Y., Teng, Z., and Zhang, Y. (2016). Expectation-regulated neural model for event mention extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 400–410. Association for Computational Linguistics.

Chang, K.-W., Samdani, R., and Roth, D. (2013). A constrained latent variable model for coreference resolution. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chang, K.-W., Samdani, R., Rozovskaya, A., Rizzolo, N., Sammons, M., and Roth, D. (2011). Inference protocols for coreference resolution. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, pages 40–44, Portland, Oregon, USA. Association for Computational Linguistics.

Chang, M., Ratinov, L., Roth, D., and Srikumar, V. (2008). Importance of semantic representation: Dataless classification. In *Proc. of the Conference on Artificial Intelligence (AAAI)*.

Chatman, S. B. (1980). *Story and discourse: Narrative structure in fiction and film*. Cornell University Press.

Chen, J., Zhang, Q., Liu, P., Qiu, X., and Huang, X. (2016). Implicit discourse relation detection via a deep architecture with gated relevance network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1726–1735.

Chen, Y., Xu, L., Liu, K., Zeng, D., and Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*.

Chen, Z. and Ji, H. (2009). Graph-based event coreference resolution. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing*.

Chen, Z., Ji, H., and Haralick, R. (2009). A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*.

Cheng, X. and Roth, D. (2013). Relational inference for wikification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Cheung, J. C. K., Poon, H., and Vanderwende, L. (2013). Probabilistic frame induction. *arXiv:1302.4813*.

Church, K. W. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ACL Conference on Applied Natural Language Processing*.

Clark, K. and Manning, C. D. (2015). Entity-centric coreference resolution with model stacking. In *ACL*.

Clark, K. and Manning, C. D. (2016a). Deep reinforcement learning for mention-ranking coreference models. In *EMNLP*.

Clark, K. and Manning, C. D. (2016b). Improving coreference resolution by learning entity-level distributed representations. In *ACL*.

Collins, M. (1999). *Head-driven Statistical Models for Natural Language Parsing*. PhD thesis, Computer Science Department, University of Pennsylvenia.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proc. of ICML*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Connolly, D., Burger, J. D., and Day, D. S. (1997). A machine learning approach to anaphoric reference. In *New Methods in Language Processing*.

Culotta, A., Wick, M., Hall, R., and McCallum, A. (2007). First-order probabilistic models for coreference resolution. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Cybulska, A. and Vossen, P. (2012). Using semantic relations to solve event coreference in text. In *Proceedings of the Workshop on Semantic relations*.

Danlos, L. and Gaiffe, B. (2003). Event coreference and discourse relations. *Philosophical Studies Series*.

Denis, P. and Baldridge, J. (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Durrett, G. and Klein, D. (2013). Easy victories and uphill battles in coreference resolution. In *EMNLP*.

Durrett, G. and Klein, D. (2014). A joint model for entity analysis: Coreference, typing, and linking. *TACL*.

Elkhlifi, A. and Faiz, R. (2009). Automatic annotation approach of events in news articles. *International Journal of Computing & Information Sciences*.

Feng, X., Huang, L., Tang, D., Ji, H., Qin, B., and Liu, T. (2016). A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 66–71. Association for Computational Linguistics.

Ferraro, F. and Van Durme, B. (2016). A unified bayesian model of scripts, frames and language. In *AAAI*.

Fillmore, C. J. (1976). Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*.

Finkel, J. R. and Manning, C. D. (2009). Nested named entity recognition. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Frermann, L., Titov, I., and M., P. (2014). A hierarchical bayesian model for unsupervised induction of script knowledge. In *EACL*.

Gabrilovich, E. and Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Int. Res.*

Ghaeini, R., Fern, X., Huang, L., and Tadepalli, P. (2016). Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 369–373. Association for Computational Linguistics.

Goldberg, Y. and Elhadad, M. (2010). An efficient algorithm for easy-first non-directional dependency parsing. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Gopal, S. and Yang, Y. (2013). Recursive regularization for large-scale classification with hierarchical and graphical dependencies.

Granroth-Wilding, M., Clark, S., Llano, M. T., Hepworth, R., Colton, S., Gow, J., Charnley, J., Lavrač, N., Žnidaršič, M., and Perovšek, M. (2015). What happens next? event prediction using a compositional neural network model.

Gutmann, M. and Hyvarinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*.

He, H., Balakrishnan, A., Eric, M., and Liang, P. (2017). Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings.

Hirschman, L., Light, M., Breck, E., and Burger, J. (1999). Deep read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

Hobbs, J. R. (1978). Resolving pronoun references. *Lingua*.

Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., and Zhu, Q. (2011). Using cross-entity inference to improve event extraction. In *ACL*.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: The 90% solution. In *Proceedings of HLT/NAACL*.

Hovy, E., Mitamura, T., Verdejo, F., Araki, J., and Philpot, A. (2013). Events are not simple: Identity, non-identity, and quasi-identity. In *NAACL-HLT*.

Hsi, A., Carbonell, J., and Yang, Y. (2015). Modeling event extraction via multilingual data sources. In *TAC*.

Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C. R., Han, J., and Sil, A. (2016). Liberal event extraction and event schema induction. In *ACL*.

Huang, R. and Riloff, E. (2012a). Bootstrapped training of event extraction classifiers. In *EACL*.

Huang, R. and Riloff, E. (2012b). Modeling textual cohesion for event extraction. In *AAAI*.

Humphreys, K., Gaizauskas, R., and Azzam, S. (1997). Event coreference for information extraction. In *Proceedings of Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.

Irwin, J., Komachi, M., and Matsumoto, Y. (2011). Narrative schema as world knowledge for coreference resolution. In *CoNLL Shared Task*, pages 86–92.

Jans, B., Bethard, S., Vulić, I., and Moens, M. F. (2012). Skip n-grams and ranking functions for predicting script events. In *EACL*, pages 336–344.

Ji, H. and Grishman, R. (2008). Refining event extraction through cross-document inference. In *ACL*, pages 254–262.

Ji, Y. and Eisenstein, J. (2015). One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, 3:329–344.

Ji, Y., Haffari, G., and Eisenstein, J. (2016). A latent variable recurrent neural network for discourse relation language models. In *Proceedings of NAACL-HLT*, pages 332–342.

Ji, Y., Tan, C., Martschat, S., Choi, Y., and Smith, N. A. (2017). Dynamic entity representations in neural language models.

Jia, R. and Liang, P. (2016). Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.

Jurafsky, D. and Martin, J. (2007). *SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*.

Kingsbury, P. and Palmer, M. (2002). From Treebank to PropBank. In *Proceedings of LREC-2002*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *ICASSP*.

Kummerfeld, J. K. and Klein, D. (2013). Error-driven analysis of challenges in coreference resolution. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Lan, M., Xu, Y., and Niu, Z. (2013). Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 476–485.

Land, A. H. and Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520.

Lappin, S. and Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational linguistics*.

Lassalle, E. and Denis, P. (2015). Joint anaphoricity detection and coreference resolution with constrained latent structures.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the CoNLL-2011 Shared Task*.

Lee, H., Recasens, M., Chang, A., Surdeanu, M., and Jurafsky, D. (2012). Joint entity and event coreference resolution across documents. In *EMNLP*.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Levy, O. and Goldberg, Y. (2014a). Dependencybased word embeddings. In *ACL*.

Levy, O. and Goldberg, Y. (2014b). Linguistic regularities in sparse and explicit word representations. In *Proc. of CoNLL*.

Li, Q., Ji, H., and Huang, L. (2013). Joint event extraction via structured prediction with global features. In *ACL*.

Liang, P. (2005). Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Liao, S. and Grishman, R. (2010). Using document level cross-event inference to improve event extraction. In *ACL*.

Lin, Z., Kan, M.-Y., and Ng, H. T. (2009). Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 343–351. Association for Computational Linguistics.

Liu, B., Hu, M., and Cheng, J. (2005). Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005*, pages 342–351.

Liu, S., Chen, Y., Liu, K., and Zhao, J. (2017). Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798. Association for Computational Linguistics.

Liu, Y. and Li, S. (2016). Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1233.

Liu, Y., Li, S., Zhang, X., and Sui, Z. (2016). Implicit discourse relation classification via multi-task neural networks. In *AAAI*, pages 2750–2756.

Liu, Z., Mitamura, T., and Hovy, E. (2015). Evaluation algorithms for event nugget detection: A pilot study. In *Proceedings of the Workshop on Events at the NAACL-HLT*.

Lu, J. and Ng, V. (2017). Joint learning for event coreference resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–101. Association for Computational Linguistics.

Luo, X. (2005). On coreference resolution performance metrics. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Marchand, H., Martin, A., Weismantel, R., and Wolsey, L. (2002). Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123(1-3):397–446.

Mausam, Schmitz, M., Bart, R., Soderland, S., and Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. In *CoNLL*.

Mihaylov, T. and Frank, A. (2016). Discourse relation sense classification using cross-argument semantic similarity based on word embeddings. In *CoNLL Shared Task*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Mitamura, T., Yamakawa, Y., Holm, S., Song, Z., Bies, A., Kulick, S., and Strassel, S. (2015). Event nugget annotation: Processes and issues. In *Proceedings of the Workshop on Events at NAACL-HLT*.

Mnih, A. and Hinton, G. (2007). Three new graphical models for statistical language modelling. In *ICML*, pages 641–648.

Mnih, A. and Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*.

Modi, A. and Titov, I. (2014). Inducing neural models of script knowledge. In *CoNLL*.

Modi, A., Titov, I., Demberg, V., Sayeed, A., and Pinkal, M. (2017). Modeling semantic expectation: Using script knowledge for referent prediction. *TACL*.

Mooney, R. J. and DeJong, G. (1985). Learning schemata for natural language processing. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence, IJCAI 1985*, pages 681–687.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. F. (2016). A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies NAACL HLT 2016*, pages 839–849.

Mostafazadeh, N., Roth, M., Louis, A., Chambers, N., and Allen, J. (2017). Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51.

Naughton, M. (2009). *Sentence Level Event Detection and Coreference Resolution*. PhD thesis, National University of Ireland, Dublin.

Ng, V. (2004). Learning noun phrase anaphoricity to improve conference resolution: Issues in representation and optimization. In *ACL*.

Ng, V. and Cardie, C. (2002a). Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *COLING*.

Ng, V. and Cardie, C. (2002b). Improving machine learning approaches to coreference resolution. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Nguyen, K., Tannier, X., Ferret, O., and Besançon, R. (2015). Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 188–197.

Nguyen, T. H., Cho, K., and Grishman, R. (2016). Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309. Association for Computational Linguistics.

Nguyen, T. H. and Grishman, R. (2016). Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 886–891. Association for Computational Linguistics.

NIST (2005). The ACE evaluation plan.

NIST, U. (2004). The ace evaluation plan. *US National Institute for Standards and Technology (NIST)*.

Paul, B., Phil, B., and Hieu, H. (2014). Oxlm: A neural language modelling framework for machine translation. *The Prague Bulletin of Mathematical Linguistics*, 102(1):81–92.

Peng, H., Chang, K.-W., and Roth, D. (2015a). A joint framework for coreference resolution and mention head detection. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, page 10, University of Illinois, Urbana-Champaign, Urbana, IL, 61801. ACL.

Peng, H., Chaturvedi, S., and Roth, D. (2017). A joint model for semantic sequences: Frames, entities, sentiments. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Peng, H., Khashabi, D., and Roth, D. (2015b). Solving hard coreference problems. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Peng, H. and Roth, D. (2016). Two discourse driven language models for semantics. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Peng, H., Song, Y., and Roth, D. (2016). Event detection and co-reference with minimal supervision. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. S. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Pichotta, K. and Mooney, R. J. (2014). Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 220–229.

Pichotta, K. and Mooney, R. J. (2016). Learning statistical scripts with lstm recurrent neural networks. In *AAAI*.

Pitler, E., Louis, A., and Nenkova, A. (2009). Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 683–691. Association for Computational Linguistics.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., and Xue, N. (2011). Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *CoNLL*.

Pradhan, S., Ramshaw, L., Weischedel, R., MacBride, J., and Micciulla, L. (2007). Unrestricted coreference: Identifying entities and events in ontonotes. In *ICSC*.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B. (2008). The penn discourse treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

Punyakanok, V. and Roth, D. (2001). The use of classifiers in sequential inference. In *Proc. of the Conference on Neural Information Processing Systems (NIPS)*, pages 995–1001. MIT Press.

Punyakanok, V., Roth, D., Yih, W., and Zimak, D. (2004). Semantic role labeling via integer linear programming inference. In *Proc. of the International Conference on Computational Linguistics (COLING)*, pages 1346–1352, Geneva, Switzerland.

Qin, L., Zhang, Z., and Zhao, H. (2016a). Shallow discourse parsing using convolutional neural network. *Proceedings of the CoNLL-16 shared task*, pages 70–77.

Qin, L., Zhang, Z., and Zhao, H. (2016b). A stacking gated neural architecture for implicit discourse relation classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2263–2270.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*.

Rahman, A. and Ng, V. (2011). Coreference resolution with world knowledge. In *ACL*.

Rahman, A. and Ng, V. (2012). Resolving complex cases of definite pronouns: the winograd schema challenge. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ramshaw, L. A. and Marcus, M. P. (1995). Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Ratinov, L. and Roth, D. (2012). Learning-based multi-sieve co-reference resolution with knowledge. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ratinov, L., Roth, D., Downey, D., and Anderson, M. (2011). Local and global algorithms for disambiguation to wikipedia. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Recasens, M., de Marneffe, M.-C., and Potts, C. (2013). The life and death of discourse entities: Identifying singleton mentions. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Ren, X., He, W., Qu, M., Huang, L., Ji, H., and Han, J. (2016). Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *EMNLP*.

Richardson, M., Burges, C. J. C., and Renshaw, E. (2013). Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 193–203.

Ritter, A., Cherry, C., and Dolan, W. B. (2011). Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.

Roth, D. and Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. In Ng, H. T. and Riloff, E., editors, *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. Association for Computational Linguistics.

Roth, D. and Zelenko, D. (1998). Part of speech tagging using a network of linear separators. In *Coling-Acl, The 17th International Conference on Computational Linguistics*, pages 1136–1142.

Rudinger, R., Rastogi, P., Ferraro, F., and Van Durme, B. (2015). Script induction as language modeling. In *EMNLP*.

Rutherford, A. and Xue, N. (2015). Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 799–808.

Sammons, M., Peng, H., Song, Y., Upadhyay, S., Tsai, C.-T., Reddy, P., Roy, S., and Roth, D. (2015). Illinois ccg tac 2015 event nugget, entity discovery and linking, and slot filler validation systems. In *Proc. of the Text Analysis Conference (TAC)*.

Sarawagi, S. and Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*.

Schank, R. C. and Abelson, R. P. (1977). Scripts, plans, goals, and understanding: An inquiry into human knowledge structures. In *JMZ*.

Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.

Schuler, K. K. (2005). Verbnet: A broad-coverage, comprehensive verb lexicon.

Song, Y., Jiang, J., Zhao, W. X., Li, S., and Wang, H. (2012). Joint learning for coreference resolution with markov logic. In *EMNLP*.

Song, Y., Peng, H., Kordjamshidi, P., Sammons, M., and Roth, D. (2015). Improving a pipeline architecture for shallow discourse parsing. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Song, Y. and Roth, D. (2014). On dataless hierarchical text classification. In *Proc. of the Conference on Artificial Intelligence (AAAI)*.

Song, Y. and Roth, D. (2015). Unsupervised sparse vector densification for short text similarity. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*

Stevenson, A. (2010). *Oxford dictionary of English*. Oxford University Press, USA.

Stolcke, A. (2002). Srilm-an extensible language modeling toolkit. In *INTERSPEECH*, volume 2002, page 2002.

Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., Buttler, D., and Hysom, D. (2010). Coreference resolution with reconcile. In *ACL*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Tilk, O., Demberg, V., Sayeed, A., Klakow, D., and Thater, S. (2016). Event participant modelling with neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 171–182. Association for Computational Linguistics.

Titov, I. and Khoddam, E. (2015). Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2015*, pages 1–10.

Tsai, C.-T., Mayhew, S., Peng, H., Sammons, M., Mangipundi, B., Reddy, P., and Roth, D. (2016). Illinois ccg entity discovery and linking, event nugget detection and co-reference, and slot filler validation systems for tac 2016. In *Text Analysis Conference (Proc. of the Text Analysis Conference (TAC) 2016)*.

Turian, J., Ratinov, L.-A., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *ACL*, pages 384–394.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*.

Wellner, B. and Pustejovsky, J. (2007). Automatically identifying the arguments of discourse connectives. In *Proceedings of the 2007 Joint Conference of EMNLP-CoNLL*.

Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005a). Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on interactive demonstrations*. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005b). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP 2005*, pages 347–354.

Winograd, T. (1972). Understanding natural language. *Cognitive psychology.*

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning global features for coreference resolution. In *NAACL.*

Wiseman, S. J., Rush, A. M., Shieber, S. M., and Weston, J. (2015). Learning anaphoricity and antecedent ranking features for coreference resolution. In *ACL.*

Xue, N., Ng, H. T., Pradhan, S., Rutherford, A., Webber, B., Wang, C., and Wang, H. (2016). Conll 2016 shared task on multilingual shallow discourse parsing. *Proceedings of the CoNLL-16 shared task*, pages 1–19.

Yang, B. and Mitchell, T. M. (2016). Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299. Association for Computational Linguistics.

Yogatama, D., Gillick, D., and Lazic, N. (2015). Embedding methods for fine grained entity type classification. In *ACL.*

Zhao, R., Do, Q., and Roth, D. (2012). A robust shallow temporal reasoning system. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).*

Zhou, Z.-M., Xu, Y., Niu, Z.-Y., Lan, M., Su, J., and Tan, C. L. (2010). Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1507–1514. Association for Computational Linguistics.

Appendix
Raw Text of Event Seeds

Type : LIFE Subtype: BEBORN

Jane Doe was born in Casper, Wyoming on March 18, 1964.

John Bobert Bond was born in England.

While investigators said they did not yet know where the Massachusetts born suspect got his guns, Scott Harshbarger, the former state attorney general who pushed for more stringent state gun control rules in the late 1990s, said, "This is where you'll see if the tracing system works."

Ali Mohammed, a native of Egypt, has admitted to five charges of conspiring with a Saudi born dissident Osama bin Laden to attack US targets in the Middle East.

For me, it's not difficult, because I was born without my hand, and I've never known any different.

They have been linked to cancer, birth defects, and other genetic abnormalities.

He calculated that Jesus' birth had occurred 532 years earlier.

Type : LIFE Subtype: MARRY

Jane and John are married.

They have been married for six years.

Ames recruited her as an informant in 1983, then married her two years later.

In 1927 she married William Gresser, a New York lawyer and musicologist.

He'd been married before and had a child.

Residents were unable to register marriages.

Type : LIFE Subtype: DIVORCE

The couple divorced four years later.

John is a divorced father of three.

He is divorced will move into guest quarters behind the presidential residence.

Jephson, who was also Prince Charles' secretary for two years, said that the Princess confided in him a

great deal, especially in the years preceding her divorce from the heir to the throne in 1996.

This year in Egypt, for example, avid campaigning helped women reverse laws that prevented them from obtaining divorce and traveling abroad without their husbands' permission.

But the Simpson trial and the jury's findings marked a turning point in the career of the twice divorced mother of two.

Type : LIFE Subtype: INJURE

Two soldiers were wounded in the attack.

The injured soldier ...

She was badly hurt in an automobile accident.

Two Palestinians were killed as they staged a drive by ambush on an Israeli jeep in the Gaza Strip near the Israeli settlement of Gush Katif Saturday afternoon, and two Israeli soldiers were wounded, one critically.

Witnesses said the soldiers responded by firing tear gas and rubber bullets, which led to ten demonstrators being injured.

Tornadoes destroyed homes and overturned cars in several areas of Alabama on Saturday and more than two dozen people were reported injured.

A fire in a bangladeshi garment factory has left at least 37 people dead and 100 hospitalized.

Type : LIFE Subtype: DIE

John Hinckley attempted to assassinate Ronald Reagan.

Terrorist groups have threatened to kill foreign hostages.

The slain leader ...

She was killed in an automobile accident.

The commander of Israeli troops in the West Bank said there was a simple goal to the helicopter assassination on Thursday of a gun-wielding local Palestinian leader.

The assassination of the once relatively obscure Fatah leader Obaiyat, whom the army blamed particularly for leading nocturnal shooting on Gilo, a neighborhood in southeastern Jerusalem, was regarded as a grave step by Israeli commentators.

The late Pope John Paul II ...

Type : MOVEMENT Subtype: TRANSPORT

The aid was aimed at repairing houses damaged by Israeli bombing and buying additional ambulances to

transport the rising number of wounded.

Zone escaped the incident with minor injuries, and Kimes was moved to the prison's disciplinary housing unit, the authorities said.

The Palestinian leaders also warned that Israel must remove its soldiers from the outskirts of Palestinian cities.

Mr. Erekat is due to travel to Washington to meet with US Secretary of State Madeleine Albright and other US officials attempting to win a ceasefire.

The weapons were moved to a secure site in the south.

Type : TRANSACTION Subtype: TRANSFER-OWNERSHIP

China has purchased two nuclear submarines from Russia.

This report concerns China's recently acquired submarines.

If the man accused of killing seven people near Boston on Tuesday got his guns in Massachusetts, he was able to skirt some of the strictest regulations in the country, people familiar with the state's laws said Wednesday.

The state requires a permit, formally known as a "firearm identification card," for purchase of virtually every kind of firearm, whether for personal protection or hunting.

There is also a scandal that erupted over Russia's declaration that it will sell weapons to Iran, contrary to the earlier made agreement.

The head of the agency's coordination program in Amman, Maher Nasser, said in a press conference that the aid was aimed at "providing food and medical aid to Palestinian refugees in the West Bank and Gaza suffering as a result of the Israeli blockade of the Palestinian Territories, as well as at repairing houses damaged by Israeli bombing and buying additional ambulances" to transport the rising number of wounded.

Type : TRANSACTION Subtype: TRANSFER-MONEY

The charity was suspected of giving money to Al Qaeda.

The organization survives on donations.

The organization is living on borrowed funds.

Actors and singers also on the flight held a benefit concert in Baghdad Saturday evening, with most of the $13 cover charge to be donated to support the Palestinian uprising.

"I'd like to see them accept his offer," said Jean Dolan, 59, a retired singing instructor who borrowed about $10,500 to buy Eircom shares in the IPO in July 1999.

Type : BUSINESS Subtype: START-ORG

Joseph Conrad Parkhurst, who founded the motorcycle magazine Cycle World in 1962, has died.

British Airways PLC plans to sell Go, its profitable cut-price subsidiary launched two years ago, the company said Monday.

We have done about 10 days of solid work across Michigan. Workers fighting for right to organize.

Type : BUSINESS Subtype: MERGE-ORG

In July, Bank of America said it planned to cut as many as 10,000 jobs as it changes its focus from growing through mergers to becoming more profitable through use of technology and operating efficiency.

Three U.S. banks, Chase Manhattan and its merger partner J.P. Morgan and Citibank, which was involved in moving about $100 million for Raul Salinas de Gortari, brother of a former Mexican president, to banks in Switzerland, are also expected to sign on, according to UBS.

Talks on a long-planned merger with KLM Royal Dutch Airlines collapsed in September.

In a drastic measure earlier this month, government controlled creditor banks named 52 financially weak companies that should be shut down or merged for sale.

Parkhurst later merged with another company that owned Road & Track to become Bond/Parkhurst Publishing.

The drug companies passed the final regulatory hurdle to their $72 billion merger, which will create the world's largest pharmaceutical company.

Type : BUSINESS Subtype: DECLARE-BANKRUPTCY

Orange County had previously filed Chapter 11 in 1995.

The bankrupt MCI-Worldcom ...

Southern California Edison says it may have to file for bankruptcy unless government officials can offer some relief.

In April of last year, the CR Company began bankruptcy procedures and the debt compensation rate of all its assets was only 5%.

Type : BUSINESS Subtype: END-ORG

The company folded in 2002.

Type : CONFLICT Subtype: ATTACK

U.S. forces continued to bomb Fallujah.

A car bomb exploded in central Baghdad.

Another exchange of gunfire in Gilo ...

Sunday night's clashes ...

... a possible coup.

Israel retaliated with rocket attacks and terrorists blew a hole in a United States warship in Yemen.

A car bomb exploded Thursday in a crowded outdoor market in the heart of Jerusalem, killing at least two people, police said.

Men in civilian clothes in the crowd began firing with AK-47 assault rifles and a 45-minute gun battle broke out.

A number of demonstrators threw stones and empty bottles at Israeli soldiers positioned near a Jewish holy site at the town's entrance.

Type : CONFLICT Subtype: DEMONSTRATE

Thousands of people rioted in Port-au-Prince, Haiti over the weekend.

The union began its strike on Monday.

Protesters rallied on the White House lawn.

The rioting crowd broke windows and overturned cars.

A crowd of 1 million demonstrated Saturday in the capital, San'a, protesting against Israel, the United States and Arab leaders regarded as too soft on Israel.

In Ramallah, around 500 people took to the town's streets chanting slogans denouncing the summit and calling on Palestinian Authority leader Yasser Arafat not to take part in it.

For weeks Italian Jewish groups, World War II veterans and leftist political parties have staged protests against a meeting between the pope and Haider, arguing that a papal encounter would lend the Austrian politician legitimacy.

More than 40,000 workers were back at their jobs Thursday following a 1-day walkout that closed social welfare offices and crippled public medical services. During the work stoppage Wednesday, local residents were unable to register marriages or get documents for real estate transactions.

Type : CONTACT Subtype: MEET

Bush and Putin met earlier this week to discuss Chechnya.

China, Japan, the United States, and both Koreas will hold a meeting this month.

Seven Lebanese Druze representatives out of eight who met under the leadership of representative Walid Jumblatt called on "youths in our Islamic Arab faction to actively join the heroic Palestinian Intifada against Israeli occupation and its agents, and to expose its means and methods."

Only representative Talal Arslan did not attend the meeting.

Tommy would again be summoned to meet prosecutors on Wednesday.

Owens complained to Defense Secretary William Cohen, prompting a meeting Friday between the governor and Gen. John Coburn, commander of the Army Material Command.

Mr. Erekat is due to travel to Washington to meet with US Secretary of State Madeleine Albright and other US officials attempting to win a ceasefire. Mrs. Albright has already met with Israel's acting foreign minister.

Egypt condemned Israel's attacks today and said it has the approval of other Arab states to host a summit with the Palestinians and Israelis only if the violence stops.

Eyewitnesses reported that Palestinians demonstrated today Sunday in the West Bank against the Sharmel-Sheikh summit to be held in Egypt tomorrow Monday.

Type : CONTACT Subtype: PHONE-WRITE

John sent an email to Jane.

All three parties discussed the matter in a teleconference Thursday.

John called Jane last night.

All else being equal, Duane Roelands would prefer to dash off short instant text messages to co-workers and friends with the service offered by Microsoft.

People can communicate with international friends without the hefty phone bills.

Unlike the telephone, people can discreetly interact with others or decide not to.

Type : PERSONELL Subtype: START-POSITION

Foo Corp. hired Mary Smith in June 1998.

Mary Smith joined Foo Corp. in June 1998.

Bill Clinton started office on January 20, 1993.

In 1997, the company hired John D. Idol to take over as chief executive.

An issue in that strike is a management plan to hire more part-time drivers and limit overtime pay.

The question of which party controls the Texas Senate is especially important this year because the Senate

will redraw congressional and legislative districts and could elect the next lieutenant governor if Gov. George W. Bush is elected president and is succeeded by Lt. Gov. Rick Perry.

Type : PERSONELL Subtype: END-POSITION

Georgia fired football coach Jim Donnan Monday after a disappointing 7-4 season that started with the Bulldogs ranked No. 10 and picked to win the SEC East, his players said.

Richard Jr. had 14 months, before he was laid off in October.

Type : PERSONELL Subtype: NOMINATE

The president nominated Rep. Mark Foley (R-Fla.) to head the commission.

The recently nominated Foley said ...

Presidential elections, including the one just ended, routinely include discussions on how a new chief executive's nominations to fill vacant positions might change the court's philosophical bent.

One of those difficult-to-dislodge judges was John Marshall, nominated by Adams to be chief justice.

Gore holds a degree from the university, and is one of about 500 people nominated for the job.

Type : PERSONELL Subtype: ELECT

Greg Lashutka was elected mayor of Columbus in 1993.

The newly elected mayor ...

Shareholders elected Sheila Johnson to a second term on the Board of Directors.

"We have a strong interest in supporting Yugoslavia's newly elected leaders as they work to build a truly democratic society," Clinton said.

The question of which party controls the Texas Senate is especially important this year because the Senate will redraw congressional and legislative districts and could elect the next lieutenant governor if Gov. George W. Bush is elected president and is succeeded by Lt. Gov. Rick Perry.

Many other Israelis have turned away from the man they elected just 18 months ago.

There is an election dispute, a tie vote as a matter of fact.

Type : JUSTICE Subtype: ARREST-JAIL

Since May, Russia has jailed over 20 suspected terrorists without a trial.

The jailed suspects demanded to speak to a lawyer.

... where Pope is incarcerated.

Asked what he had done to attract attention since he was incarcerated, Chapman recalled a 1987 interview with People magazine, for which he received $5,000, according to news reports at the time.

Abu Talb, the last major prosecution witness, has been jailed in Sweden for attacks against Jewish and American targets in Europe.

The youngest son of ex-dictator Suharto disobeyed a summons to surrender himself to prosecutors on Monday and be imprisoned for corruption.

Florida police arrested James Harvey in Coral Springs on Friday.

A judicial source said today, Friday, that five Croatians were arrested last Tuesday during an operation targeting a number of war criminals suspected of involvement in the killing of around 50 Serbian civilians in 1991.

A court of appeals on Tuesday suspended Gen. Augusto Pinochet's house arrest while it studied a judge's explanation for indicting the former dictator on homicide and kidnapping charges.

Canadian authorities arrested two Vancouver area men on Friday and charged them in the deaths of 329 passengers and crew members of an Air India Boeing 747 that blew up over the Irish Sea in 1985, en-route from Canada to London.

Type : JUSTICE Subtype: RELEASE-PAROLE

Harvey was released the following day.

The newly freed prisoners ...

Russian President Vladimir Putin says he will pardon and release American businessman Edmond Pope.

Type : JUSTICE Subtype: TRIAL-HEARING

Jenna Raleigh will be tried in a military court.

Clinton also touched on the matter of American Edmond Pope who is being tried in a closed court in Russia on charges of spying.

And so the case is being tried in federal court, where prosecutors can, and say they will, seek the death penalty.

A Palestinian terrorist began his testimony Friday in the trial of two Libyans accused of bombing Pan Am Flight 103, describing his role in attacks against Israel in the 1970s.

The trial resumed this week after a month of delays following the disclosure that new evidence surfaced on another group, the Damascus based Palestinian Front for the Liberation of Palestine General Command.

Clinton also touched on the matter of American Edmond Bob who is being tried in a closed court in

Russia on charges of spying.

Stewart's hearing will be held on Monday in the superior court.

Midway through the hearing, Chief Justice Renquist seemed to scold his colleagues for being too talkative when he made an unusual offer to the lawyer representing Florida's Attorney General.

At a preliminary hearing Friday afternoon, Sauls made it clear he would take a no nonsense approach to the trial.

Type : JUSTICE Subtype: CHARGE-INDICT

Joy Fenter was indicted by a grand jury on eleven counts of mail fraud.

Milosevic, who has been indicted by the international war crimes tribunal in The Hague, Netherlands, cannot leave Yugoslavia without risking arrest and extradition.

Guzman indicted Pinochet, holding him responsible for the actions by the "Caravan of Death," a military party that killed 73 political prisoners shortly after the 1973 coup in which Pinochet ousted Marxist President Salvador Allende.

In an eight-count indictment, the men were charged with using suitcases packed with explosives to bomb two Air India jets on the same day, June 23, 1985.

Ryan Mathers was charged with reckless endangerment.

Appointed to the federal bench in 1979, he was charged two years later with conspiracy to accept a bribe in a case he presided over in Miami.

Bagri was also charged with trying to murder Tara Singh Hayer, editor of The Indo-Canadian Times, North America's largest Punjabi newspaper, in 1998.

Type : JUSTICE Subtype: SUE

Donald Crutchfield filed suit against Toys 'R' Us in 1997.

Five years there, $30 million. U.S. victims of terrorism have been able to sue foreign governments since 1996.

Brentwood Academy responded with a lawsuit that has made its way to the U.S. Supreme Court, where arguments will be made Wednesday.

Type : JUSTICE Subtype: CONVICT

Martha Breckenridge was convicted of two counts of manslaughter.

Tommy, a multimillionaire with a playboy image and love of fast cars, is the first member of Suharto's

family to be convicted of graft.

It found him guilty of enriching himself through a property deal with the state's main food supply agency.

A Russian court convicted Pope Wednesday on espionage charges and sentenced him to 20 years in prison.

Type : JUSTICE Subtype: SENTENCE

She was sentenced to life without parole.

Hutomo "Tommy" Mandala Putra, 37, was sentenced to 18 months in prison on Sept. 22 by the Supreme Court, which overturned an earlier acquittal by a lower court.

A Russian court convicted Pope Wednesday on espionage charges and sentenced him to 20 years in prison.

46-year-old Abu Talib was sentenced to life imprisonment in 1990 in Sweden for terrorist acts in Amsterdam, Copenhagen and Stockholm between 1985 and 1986.

Solomon could be sentenced to up to 211 years in prison.

Type : JUSTICE Subtype: FINE

Ms. Brooks, who could go to prison and will certainly be heavily fined has agreed to turn state's evidence, turning against her boss.

It fined the school $3,000 and banned its football program.

The company was ordered to pay a fine of $300,000.

Type : JUSTICE Subtype: EXECUTE

David Goran was executed by lethal injection in March 1987.

Twelve executed prisoners have been posthumously exonerated.

She recently sold the film rights to her latest book, "Saints and Villains," about Dietrich Bonhoeffer, the German theologian executed by the Nazis for plotting against Hitler.

Bush said he might change his mind if he did not think that executions saved lives.

Type : JUSTICE Subtype: EXTRADITE

The former leader was extradited to Burkina Faso.

Milosevic, who has been indicted by the international war crimes tribunal in The Hague, Netherlands, cannot leave Yugoslavia without risking arrest and extradition.

"In the end, Milosevic may even prefer extradition to The Hague rather than stay here and face our justice," said opposition leader Zarko Korac.

Kimes' main demand was that his mother not be extradited to California, where the two face the death penalty on charges they killed a former business associate.

Type : JUSTICE Subtype: ACQUIT

Chase was acquitted after a trial in the Senate.

He was acquitted by a jury in 1983, but a panel of judges reopened the case four years later, accusing him of both the original crime and lying about it under oath.

Type : JUSTICE Subtype: APPEAL

Defense attorneys said they will appeal.

Type : JUSTICE Subtype: PARDON

Russian President Vladimir Putin says he will pardon and release American businessman Edmond Pope.