



AFRL-RI-RS-TR-2018-101

DETECTING RELATIONS AND ANOMALY IN TEXT AND SPEECH (DRATS)

COLUMBIA UNIVERSITY

APRIL 2018

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

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

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

AFRL-RI-RS-TR-2018-101 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

WALTER V. GADZ JR.
Work Unit Manager

/ S /

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

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

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188				
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.								
1. REPORT DATE (DD-MM-YYYY) APR 2018		2. REPORT TYPE FINAL TECHNICAL REPORT		3. DATES COVERED (From - To) OCT 2012 – NOV 2017				
4. TITLE AND SUBTITLE DETECTING RELATIONS AND ANOMALY IN TEXT AND SPEECH (DRATS)				5a. CONTRACT NUMBER FA8750-12-2-0347				
				5b. GRANT NUMBER N/A				
				5c. PROGRAM ELEMENT NUMBER 62303E				
6. AUTHOR(S) Owen Rambow, Mona Diab, Julia Hirschberg, Kathleen McKeown, Smaranda Muresan, Mari Ostendorf				5d. PROJECT NUMBER DEFT				
				5e. TASK NUMBER 12				
				5f. WORK UNIT NUMBER 03				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <table style="width: 100%; border: none;"> <tr> <td style="width: 33%; vertical-align: top;"> <u>Prime:</u> Columbia University 116TH Street and Broadway New York, NY 10019 </td> <td style="width: 33%; vertical-align: top;"> <u>Sub:</u> George Washington Univ. 2121 Eye Street, NW Washington DC 20052 </td> <td style="width: 33%; vertical-align: top;"> <u>Sub:</u> Univ. of Washington Seattle, WA 98195 </td> </tr> </table>				<u>Prime:</u> Columbia University 116 TH Street and Broadway New York, NY 10019	<u>Sub:</u> George Washington Univ. 2121 Eye Street, NW Washington DC 20052	<u>Sub:</u> Univ. of Washington Seattle, WA 98195	8. PERFORMING ORGANIZATION REPORT NUMBER	
<u>Prime:</u> Columbia University 116 TH Street and Broadway New York, NY 10019	<u>Sub:</u> George Washington Univ. 2121 Eye Street, NW Washington DC 20052	<u>Sub:</u> Univ. of Washington Seattle, WA 98195						
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RIEA 525 Brooks Road Rome NY 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI				
				11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2018-101				
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09								
13. SUPPLEMENTARY NOTES								
14. ABSTRACT Our project is called DRATS (Detecting Relations and Anomaly in Text and Speech). Most of our tasks have revolved around an analysis of the human participants in communications and in reported events. Specifically, we have concentrated on two areas of investigation. Cognitive State: What is going on in people's minds? We have concentrated specifically on belief and sentiment: who believes what, with what degree of certitude, and who has what sentiment towards what? Cognitive Relations: How are people related to each other? We have investigated power, influence, community endorsement, and social networks, in general. In addition, we have also addressed some core information extraction tasks.								
15. SUBJECT TERMS Relation Extraction, anomaly, knowledge base, belief and sentiment detection, sarcasm detection, hedging								
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 148	19a. NAME OF RESPONSIBLE PERSON WALTER V. GADZ JR.			
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code)			

Contents

1	Summary of Effort	1
1.1	Overview	1
1.2	Our Tasks	1
1.3	Our Research Activities.....	1
1.4	This Document	2
I	Core Information Extraction	3
2	Information Extraction	4
2.1	Summary	4
3	Named Entity Recognition (NER)	5
3.1	Introduction	5
3.1.1	English NER Systems	5
3.1.2	Arabic NER Systems	6
3.1.3	NER for Low Resource Languages	7
3.2	NER for Dialectal Arabic (DA-NER)	7
3.2.1	Methods, Assumptions, and Procedures	7
3.2.2	Results and Discussion	10
3.3	Cross-Lingual Transfer Learning NER	12
3.3.1	Introduction.....	12
3.3.2	Methods, Assumptions, and Procedures	13
3.3.3	Results and Discussion	16
3.4	Transfer Learning MSA-DA NER	16
3.4.1	Introduction.....	16
3.4.2	Methods, Assumptions, and Procedures	17
3.4.3	Results and Discussion	18
3.5	Deep Neural Networks DA-NER Model using Multi-representation Embeddings.....	19
3.5.1	Introduction.....	19
3.5.2	Methods, Assumptions, and Procedures	19
3.5.3	Results and Discussion	21
3.6	Conclusions	24
4	Entity Linking	26
4.1	Introduction	26
4.2	Methods, Assumptions, and Procedures.....	26
4.2.1	Datasets & Tools	28

4.2.2	NIL Clustering approach	29
4.3	Results and Discussion	29
4.4	Conclusions	30
5	Named Entity Alias Resolution (NEAR)	31
5.1	Introduction	31
5.2	Methods, Assumptions, and Procedures	31
5.2.1	Current Aliasing Resolution Approach	31
5.2.2	Datasets & Tools	33
5.3	Results and Discussion	34
5.4	Conclusions	36
II	Cognitive State	37
6	Belief Word Tagging	38
6.1	Summary	38
6.2	Introduction	38
6.2.1	Background	39
6.3	Methods, Assumptions, and Procedures	39
6.3.1	Approach	39
6.3.2	Level of Committed Belief on Proposition (LCB-P)	39
6.3.3	Multi Task learning for LCB-E and WSC	40
6.3.4	Experiments and Evaluation	41
6.4	Results and Discussion	42
6.5	Conclusions	42
7	Sentiment	44
7.1	Summary	44
7.2	Sentiment Analysis for Social Media	44
7.2.1	Introduction	44
7.2.2	Methods, Assumptions, and Procedures	44
7.2.3	Results and Discussion	45
7.2.4	Conclusions	45
7.3	Sentiment Analysis for Spanish Twitter	45
7.3.1	Introduction	45
7.3.2	Methods, Assumptions, and Procedures	46
7.3.3	Results and Discussion	46
7.3.4	Conclusions	46
7.4	Sentiment Analysis using Word Embeddings	46
7.4.1	Introduction	46
7.4.2	Methods, Assumptions, and Procedures	46
7.4.3	Results and Discussion	47
7.4.4	Conclusions	47

8	Source and Target Belief and Sentiment	48
8.1	Summary	48
8.2	Introduction	48
8.2.1	Data.....	48
8.3	English Sentiment 1.....	49
8.3.1	Methods, Assumptions, and Procedures	49
8.3.2	Results and Discussion	50
8.3.3	Conclusions.....	50
8.4	English Sentiment 2.....	51
8.4.1	Methods, Assumptions, and Procedures	51
8.4.2	Data Pre-Processing.....	51
8.4.3	Approach.....	51
8.4.4	Results and Discussion	51
8.4.5	Conclusions.....	52
8.5	Chinese Sentiment.....	52
8.5.1	Methods, Assumptions, and Procedures	52
8.5.2	Results and Discussion	52
8.5.3	Conclusions.....	53
8.6	Spanish Sentiment	53
8.6.1	Methods, Assumptions, and Procedures	53
8.6.2	Results and Discussion	54
8.6.3	Conclusions.....	54
8.7	English Belief	54
8.7.1	Methods, Assumptions, and Procedures	54
8.7.2	Results and Discussion	55
8.7.3	Conclusions.....	55
8.8	Chinese Belief	56
8.8.1	Methods, Assumptions, and Procedures	56
8.8.2	Results and Discussion	56
8.8.3	Conclusions.....	56
8.9	Spanish Belief.....	58
8.9.1	Methods, Assumptions, and Procedures	58
8.9.2	Results and Discussion	58
8.10	Conclusion for All Work in This Chapter.....	58
9	Identifying Hedging	59
9.1	Summary	59
9.2	Introduction	59
9.3	Methods, Assumptions, and Procedures.....	59
9.3.1	Identifying Hedge Terms	60
9.3.2	Hedge Features	61
9.4	Results and Discussion.....	61
9.4.1	Experiments and Results.....	61
9.5	Conclusions	62

10	Sarcasm Detection	63
10.1	Summary	63
10.2	Sarcasm Detection as a Word-Sense Disambiguating Problem	63
10.2.1	Introduction.....	63
10.2.2	Methods, Assumptions and Procedures	63
10.2.3	Results and Discussion	64
10.2.4	Conclusion	65
10.3	The Role of Conversation Context for Sarcasm Detection.....	66
10.3.1	Introduction.....	66
10.3.2	Methods, Assumptions and Procedures	66
10.3.3	Results and Discussion	67
10.3.4	Conclusion	68
III	Cognitive Relations	69
11	Detecting Community Endorsement	70
11.1	Summary	70
11.2	Introduction	70
11.3	Methods, Assumptions and Procedures.....	71
11.4	Results and Discussion	72
11.4.1	Resulting Resources.....	75
11.5	Conclusions	76
12	Influence	77
12.1	Summary	77
12.2	Situational Influence.....	77
12.2.1	Introduction.....	77
12.2.2	Methods, Assumptions and Procedures	78
12.2.3	Results and Discussion	84
12.2.4	Conclusions.....	85
12.3	Persuasive Influence.....	86
12.3.1	Introduction.....	86
12.3.2	Methods, Assumptions and Procedures	87
12.3.3	Results and Discussion	89
12.3.4	Conclusions.....	90
13	Power Relations	91
13.1	Summary	91
13.1.1	Goal of Work	91
13.1.2	Main Methods.....	91
13.1.3	Resulting Resources.....	91
13.2	Introduction	91
13.3	Methods, Assumptions, and Procedures.....	92
13.3.1	Motivation.....	92
13.3.2	Data.....	93
13.3.3	Problem Formulation	93
13.3.4	Structural Analysis.....	94

13.4	Results and Discussion	96
13.5	Conclusions	97
14	Social Network Extraction	98
14.1	Summary	98
14.1.1	Goal of Work	98
14.1.2	Resulting Resources.....	98
14.2	Introduction	98
14.3	Methods, Assumptions and Procedures.....	99
14.3.1	Social Events and Data	99
14.3.2	SINNET: Social Interaction Network Extractor from Text.....	99
14.3.3	Social Network Analysis Metrics	101
14.3.4	Literature Survey	101
14.4	Results and Discussion	102
14.4.1	Social Event Detection.....	102
14.4.2	Social Network Extraction.....	102
14.5	Conclusions	104
15	Argument Mining: Macro-level and Micro-level	105
15.1	Summary	105
15.2	Argumentation Features and Persuasive Essay Quality	106
15.2.1	Introduction.....	106
15.2.2	Methods, Assumptions and Procedures	106
15.2.3	Results and Discussion	107
15.2.4	Conclusions.....	107
15.3	Analyzing the Semantic Types of Claims and Premises.....	107
15.3.1	Introduction.....	107
15.3.2	Methods, Assumptions and Procedures	108
15.3.3	Results and Discussion	108
15.3.4	Conclusions.....	109
15.4	Argument Schemes.....	109
15.4.1	Introduction.....	109
15.4.2	Methods, Assumptions and Procedures	110
15.4.3	Results and Discussion	110
15.4.4	Conclusions.....	110
IV	Speech Processing	112
16	Speech Processing	113
16.1	Summary	113
16.2	Introduction	113
16.3	Methods, Assumptions and Procedures.....	114
16.4	Results and Discussion	114
16.5	Resulting Resources	115
16.6	Conclusions	116

V Technology, Resources, Deliverables, Publications	117
17 New Technology	118
17.1 Deliverables.....	118
17.2 ADEPT Algorithms.....	118
17.3 Knowledge Base Population and Languages.....	119
18 Publications	120
18.1 Belief and Hedging.....	120
18.2 Sentiment.....	120
18.3 Social Network Extraction	120
18.4 Power.....	120
18.5 Sarcasm	120
18.6 Community Endorsement.....	120
19 Acronyms	121

List of Figures

1	BC Data Genre and Performance correlation.....	12
2	Transfer Learning System Framework for Low resources Languages	14
3	Bi-LSTM-CRF DA-NER Model.....	19
4	LSTM Cell	20
5	Size vs. Performance in deep learning and conventional ML; adopted from Andrew Ng slides on Neural Networks and Deep Learning	22
6	Resolution System.....	27
7	Entity Clustering Example	32
8	Multi-task hierarchical bidirectional LSTM learning model for LCB and WS/genre classification.....	39
9	A pipeline to add “The author says”; convert ERE-annotated files into the proper APF (Ace Program Format) files and convert Source files to *.sgm format which is the format expected by the SINNET system.....	57
10	Example of AMT word disambiguation task	61
11	Sentence-level Attention Network for Context and Reply. Figure is inspired by (Yang et al., 2016)	67
12	Influence Example: A Portion of a Wikipedia Talk Page Discussion Regarding an Image for Death Displaying Richard001 as the Influencer. Replies Indicated by Indentation (e.g., P2 is a Response to P1)	78
13	Statistics for each corpus; LiveJournal and Wikipedia.....	79
14	Accuracy for cross-validation within each corpus and cross-domain between each corpus. We experimented training on LiveJournal and testing on Wikipedia (L-W) and training on Wikipedia and testing on LiveJournal (W-L). We use two baselines: The majority class, and the question feature. The features used are question (?), social media (sm), sentiment, belief, n-grams, and pos.	80
15	Statistics for full datasets.....	80
16	The effect, in F-score, of conversational structure in the ABCD corpus. Statistical significance is shown over majority (α) and n-gram (β) baselines.....	81
17	The effect, in F-score, of conversational structure in the IAC test set using the IAC and ABCD as training data. Results highlighted to indicate statistical significance over majority (α) and n-gram (β) baselines.	81

18	The effect, in F-score, of conversational structure in the AWTP test set using the IAC and ABCD as training data. Statistical significance is shown over majority (α) and n-gram (β) baselines.....	82
19	The size (in users) of each trait corpus	83
20	The author trait results for accuracy	83
21	List of Statistics for Each of the Annotated Influence Datasets.....	85
22	Cross-Validation Results.....	86
23	Influence Detection Results: The Results Are Shown when Training in a Single Domain (SD) and Applying Domain Adaptation (DA). Performance Is in Terms of F-score Using the Baselines (Everyone Is an Influencer, and Number of Words), All Features (Full), Individual Features One at a Time, and the Best Feature Combination for Each Data Set which Differs for each Genre. Significance Is Shown in Comparison to all-yes Baseline (α), Num-words Baseline (β), Domain Adaptation to Single Domain (γ)	87
24	Dependency parse tree for the sentence (in the ACE corpus): <i>[Toujan Faisal], 54, said [she] was informed of the refusal by an [Interior Ministry committee] overseeing election preparations.....</i>	100

List of Tables

1	Morphological Features	8
2	Twitter and DA-EGY dataset statistics.....	10
3	DA-NER Feature Set.....	11
4	Twitter DA-NER Results.....	12
5	DA-EGY NER Results.....	13
6	Transfer Learning Dataset Language-TagsNumbers Stats	16
7	F-score Results.....	17
8	MSA & MSA to DA TL performance results.....	18
9	AQMAR dataset stats	21
10	DA-NER system performance results	23
11	MSA-augmented DA-NER system performance results	24
12	Entity Type Classifier Results	28
13	Entity Linking and Clustering train and test data stats	29
14	Precision (P), Recall (R), F-score (F1) of GWU teamat TAC-KBP	30
15	F-score PNEAR results with the use of different features.....	35
16	PNEAR System performance when tested on Transliteration vs. Variant cases.....	36
17	Training and evaluation set statistics.....	41
18	Training and evaluation set statistics forWSC task	41
19	Results obtained for the LCB-P and P15 (of (Prabhakaran et al., 2015b)) model on the LDC 2014 Evaluation dataset.	42
20	Results of employing MTL network on LCB-E and WSC tasks	43
21	Results for the TAC 2014 task.....	45
22	Results in the SemEval2014 evaluation(Subtask B).....	45
23	Sentiment Analysis Results at a Global Level.....	46
24	Results Compared to other Models. The bold and underlined, and bold-only numbers are the best and second best results respectively. Our NBLR + POSwemb performs the best or second on several datasets.....	47
25	Results for our English Sentiment System-2 on “SuperDev” Data	51
26	Results for our two English Sentiment systems on KBP Eval Data, along with the baseline .	52
27	Results for our Chinese Sentiment System on the development dataset	53
28	Results for our Chinese Sentiment system on KBP Eval Data, along with the baseline.....	53
29	Results for our two Spanish Sentiment systems on TAC KBP Evaluation Data.....	54
30	Results for our three English Belief systems on KBP Eval Data, along with the official baseline	55

31	Results for our three Chinese Belief systems on KBP Eval Data, along with the official baseline. Note that no belief was predicted on the predicted ERE, neither by the official baseline, nor by our systems; we therefore omit this condition from the table.....	57
32	Results for our three Spanish Belief systems on KBP Eval Data, along with the official baseline. Note that no belief was predicted on the predicted ERE, neither by the official baseline, nor by our systems; we therefore omit this condition from the table.....	58
33	List of (potential) hedge words and phrases.....	59
34	Examples of rules used to disambiguate hedge terms	60
35	Belief results using (a) no hedge detection, (b) dictionary-based hedge detection, and (c) rule-based hedge detection.	62
36	Evaluation of classification approaches (SVMbl and kernelWe) for LSSD experiments	65
37	Experimental results for the discussion forum dataset (bold are best scores)	68
38	Experimental results for Twitter dataset (bold are best scores).....	68
39	Truncated Discussion Thread in Change My View.....	88
40	Results of Persuasive Influence Prediction Task	90
41	Data Statistics. Row 1 presents the total number of threads in different subsets of the corpus. Row 2 and 3 present the number of interacting participant pairs (<i>IPP</i>) and related interacting participant pairs (<i>RIPP</i>) in those subsets.	93
42	Student's t-Test Results of f^{pIMt} . THR^{New} : new meta-data features; THR^{PR} , DIA^{PR} : meta-data and dialog-act features from previous studies; * ($p < .05$); ** ($p < .01$); *** ($p < .001$). 95	
43	Accuracies on feature subsets (<i>dev</i> set). THR^{New} : new meta-data features; THR^{PR} , DIA^{PR} : meta-data and dialog-act features from previous studies; LEX: ngrams; BEST: best subset; IMt stands for IMt(p1, p2).....	97
44	The distribution of social events in the training and test sets used for experiments	99
45	Results for training on ACE-train and testing on Alice-test. P refers to Precision, R refers to Recall, F refers to F1-measure. Terminology used for the tree structures is explained in detail in Section 14.3.2.....	102
46	Terminology used to present results in Table 47	103
47	Results comparing the two baseline systems (B-Simple and B-BOW) with the models trained on the tree kernel structures discussed in Section 14.3.2. $\uparrow$ means greater value is better. $\downarrow$ means lesser value is better. Network density of the gold network is 0.0166. The number of connected components in the gold network are 34. The number of triads in the gold network are 103.....	103
48	Argumentation Features.....	106
49	Disfluency type detection F-score in different corpora using only the Switchboard corpus in training.	115

Chapter 1

Summary of Effort

1.1 Overview

The Columbia DARPA DEFT project is called DRATS (Detecting Relations and Anomaly in Text and Speech). Our emphasis has not been on core Information Extraction (IE) tasks: detecting who did what to whom when and where. Instead, most of our tasks have revolved around an analysis of the human participants in communications and in reported events. Specifically, we have concentrated on two areas of investigation:

- Cognitive State: What is going on in people's minds? As we communicate, we use our cognitive state to generate linguistic contributions. We have concentrated specifically on belief and sentiment: who believes what with what degree of certitude, and who has what sentiment towards what? In addition, Cognitive State affects and is affected by argumentation, which we have also researched.
- Cognitive Relations: How are people related to each other? We are fundamentally social beings, so understanding the relations among people is a major component of understanding language. We have investigated power, influence, community endorsement, and social networks in general.

In addition, we have also addressed some core IE tasks.

1.2 Our Tasks

The following are the original tasks of the Columbia-led DRATS effort. Our original tasks A1 (Interaction) and A4 (Speech Anomaly) were wound down as speech was de-emphasized in the program. We list all original tasks below, but note that for most of the program we concentrated on tasks A2, A3, and A5.

Task	Description	Duration	Sites	PIs
A1	Interaction	1-54	Columbia, UW	Hirschberg
A2	Cognitive State	1-54	Columbia, GW	Diab, Hirschberg, McKeown, Muresan, Rambow
A3	Cognitive Relations	1-54	Columbia, UW	McKeown, Ostendorf, Rambow
A4	Speech Anomaly	10-54	Columbia, UW	Ostendorf
A5	ERE, Novelty, and Conflict	1-54	Columbia, GW	Diab, Muresan

1.3 Our Research Activities

We now list the specific research activities that we are engaged in.

In column "Eval 2016", the entry "Proposed" means that we proposed a new evaluation, but this was not adopted.

DRATS Task	Activity Description	ADEPT Algo	Eval 2015	Eval 2016	Sites	PIs
A2	Belief	6.1	Belief Pilot	BeSt	Columbia, GW	Diab, Rambow
A2	Hedging				Columbia	Hirschberg
A2	Sentiment	6.5	Sent. Slot Fill.	BeSt	Columbia	McKeown
A2	Sarcasm	6.2, 6.7	—	BeSt	Columbia	Muresan
A3	Social Influence				Columbia	McKeown
A3	Social Power	6.8	Pilot	Proposed Open	Columbia	Rambow
A3	Community Endorsement	6.10	dry run	—	UW	Ostendorf
A3	Social Network Extr.	6.6	—	—	Columbia	Rambow
A5	Sentence Similarity	6.3	SemEval	SemEval	GW	Diab
A5	Argumentation Mining	6.8	—	Through 6.8	Columbia	Muresan
A5	NER and Linking	6.11	TAC-KBP	TAC-KBP	GW	Diab
A5	Event Detection	6.12	-	TAC-KBP	GW	Diab

In addition, we participated (and helped organize) the 2017 BeSt evaluation as well.

1.4 This Document

This document is structured as follows. We divide the document into four parts:

- Part I covers our work in core information extraction. This was not our emphasis, but we made some important contributions.
- Part II covers our work on modeling cognitive state: belief and sentiment.
- Part III covers our work on modeling social relations.
- Part IV deals with our work on speech processing, which we did at the beginning of the project before the project was re-oriented.

We conclude with contributions.

A better rendered pdf version of this document is available from the PI upon email request.

Part I

Core Information Extraction

Chapter 2

Information Extraction

2.1 Summary

In this part of the report, we illustrate our work on Named Entity Recognition (NER), Named Entity Linking (NEL), and Named Entity Alias Recognition (NEAR). We illustrate our approach to NER/NEAR for low resource scenarios. We started investigating event detection using the results on semantic role labeling (SRL). SRL is a relevant component for question answering as well as event detection and classification in general.

Chapter 3

Named Entity Recognition (NER)

3.1 Introduction

It has been shown that an essential component of an IE system is identifying the names of people, locations, organizations, and other named entity types (Nadeau and Sekine, 2007). Identifying references to these entities in text is also recognized as an important task of an IE system and has been shown to improve Information Retrieval performance due to the high frequency of names in text (Thompson and Dozier, 1997).

Named entity recognition (NER) has been used in many domains and tasks such as Question Answering (QA), where (Ferrández et al., 2007) shows that, on average, questions contain about 85% Named Entities. Additionally, it is an important pre-task step in many problems such as Entity Linking and Clustering, where identifying the named entities in the text is essential.

NER is a crucial component in our NEAR system, as named entities need to be identified to be linked/clustered.

A lot of work has been done in NER and it is considered reasonably mature especially for high resource languages such as English and Spanish. However, the NER performance for other languages that gained later interest or for lower-resource languages (e.g. Arabic, Hebrew, and French) is far from its counterpart of high-resource languages and is still considered a challenge. Additionally, some informal genres (e.g. microblogs, weblogs) present the challenge of using dialects (e.g. Dialectal Arabic) in less formal texts, thus we have the urgent need to build NER systems for dialectal Arabic.

In the majority of NER systems the conventional list of entity types includes person (*PER*) (e.g., Barack Obama), location (*LOC*) (e.g., Potomac River), and organization (*ORG*) (e.g., Congress). Other systems include additional entity types such as geopolitical entities *GPE* (e.g., the US), and facility *FAC* (e.g., the Berlin Wall). In this work we focus on the first three named entity types; for us, the *LOC* type includes the union of man-made locations and geopolitical entities.

This chapter is organized as follows: we start with an overview of NER preprocessing steps and evaluations, followed by a brief overview of English and Arabic NER systems. We then build up the motivation behind our DA-NER system. Then we discuss NER for low resource languages and introduce our proposed algorithm for a transfer learning NER model from high to low resource languages. The following sections describe our deep learning NER model. Finally, we extend and implement our transfer learning model to Arabic (transferring from Modern Standard Arabic to an Arabic dialect).

3.1.1 English NER Systems

A lot of work has been done in the area of English NER, where the F1-score of state-of-the-art systems is in the range of about 90%. Examples of widely used English NER systems are: Stanford NER (Finkel et al., 2005), which we will employ for English preprocessing), in addition to (Ratinov and Roth, 2009; Lample

et al., 2016b). However, it is worth noting that the majority of NER systems report results on newswire data that is characterized by well structured text. In contrast, testing the same systems on microblogs (e.g. Twitter data) or weblogs has demonstrated a dramatic decrease in performance with F1=44% when using Stanford NER on Twitter data ((Ritter et al., 2011)).

State-of-the-art NER systems use supervised machine learning approaches, whether they use conventional machine learning techniques or deep learning models, where a model is generated using training data (gold annotated data). A myriad of supervised approaches have been tested with the NER task such as Support Vector Machine (SVM) as shown in (Isozaki and Kazawa, 2002), Perceptron (Torisawa, 2007), Conditional Random Field (CRF) (McCallum and Li, 2003; Lafferty et al., 2001), and neural network models (Lample et al., 2016b; Dernoncourt et al., 2017). The selected feature set plays a critical role in the final performance results. Thus, most of the successful English NER models focus on feature engineering to maximize system performance. However, with the bloom of deep learning models and their successful results achieved for multiple NLP tasks (Collobert and Weston, 2008), recent NER systems have successfully used NN models to achieve almost state-of-the-art results (Dernoncourt et al., 2017). Common features that are found in state-of-the-art systems include, but are not limited, to:

- Contextual Features: such as the current and surrounding words within a window of size N.
- Current word character n-grams.
- Current and Surrounding Part of Speech (POS) tags.
- Current and Surrounding Word Shape. The word shape includes, but is not limited, to: *numerical*, *alpha-numerical* and *capitalization*, which provides a strong clue for named entities in English and other languages that use capitalization.
- Gazetteers: A gazetteer is a dictionary that contains a list of named entities categorized by entity type/class (e.g. PER).
- Words and character embeddings, especially for neural network models

Since highly efficient English NER systems are readily available, we opt to use a competitive off-the-shelf English NER system in our work (Finkel et al., 2005), which is part of Stanford CoreNLP toolset (Manning et al., 2014a).

3.1.2 Arabic NER Systems

Although Arabic NER systems are not as mature as English NER systems, there has been a lot of work in the area of Arabic NER. Similarly to English, Arabic NER systems mainly target well formed data in Modern Standard Arabic (MSA) such as newswire (NW) genre (Benajiba and Rosso, 2008), with few instances of NER systems that experimented with microblogs data (Darwish and Gao, 2014).

As for English, there have been recent systems that target microblogs, and systems that target dialectal data as well using character embedding neural network models. We compare our results to a previous state-of-the-art system (Darwish and Gao, 2014).

Arabic NER systems inherit the challenges imposed by Arabic, as it is a morphologically rich and highly inflected language. Some of the challenges that Arabic NER faces are (Abdul-Hamid and Darwish, 2010):

- Lack of capitalization: capitalization in Latin languages has been a strong indicator for named entities. However, in Arabic, proper nouns are not capitalized, which renders the identification of NEs more complicated;

- Some words can be proper nouns, nouns, or adjectives. For instance, *gamilah*, *jmylp*, which means ‘beautiful’, can be a proper noun or an adjective. Another example, *gamal*, *jmAl*, which means ‘beauty’, is a noun but can be a regular noun or a person name;
- Agglutination: since Arabic exhibits concatenate morphology, we note the pervasive presence of affixes agglutinating to proper nouns as prefixes and suffixes (Shaalán, 2014). For instance: Determiners appear as prefixes as in *Al* (*AlqAhrp* ‘Cairo’), likewise with affixival prepositions such as *l*, meaning ‘for’, (*Idm\$q* ‘to/from Damascus’), as well as prefixed conjunctions such as *w*, meaning ‘and’, (*wAlqds* ‘and Jerusalem’); and
- Absence of Short Vowels (diacritics): written MSA, even in newswire, is undiacritized which results in higher ambiguity that can only be resolved using contextual information (Benajiba et al., 2009). Examples of ambiguity are: *mSr*, which is underspecified for short vowels, can refer to *miSor* ‘Egypt’ or *muSir* ‘insistent’. Another example of this phenomena: *qTr* may be ‘Qatar’ if diacritized as *qaTar*, ‘sugar syrup’ if *qaTor*, ‘diameter’ if *quTor*.

In MSA-NER we focus on building a system that is fully integrated into MADAMIRA (Pasha et al., 2014) using a state-of-the-art feature set, similarly to what is proposed in (AbdelRahman et al., 2010) in a supervised machine learning approach (MADAMIRA-Cogent implementation of linear SVM classification).

3.1.3 NER for Low Resource Languages

Both English and MSA are considered high resource languages where we have a large/good size of annotated data for training. However, annotation is very expensive and not available for all languages. Thus, the need to build NER systems for low resource languages (e.g. Dialectal Arabic or DA, Chinese) without the availability of annotated data in every targeted low resource language is vital. In the following sections we are going to discuss our transfer learning NER approach. Additionally, we discuss DA-NER since DA is considered a low resource language. We introduce our supervised DA-NER approach using the available annotations, the transfer learning approach from MSA to DA as an extension for our proposed transfer learning model, and our deep learning pattern embedding NER model.

3.2 NER for Dialectal Arabic (DA-NER)

3.2.1 Methods, Assumptions, and Procedures

Arabic NER systems fail to deliver good results when tested on microblogs or any form of data that uses Dialectal Arabic (DA). As a result, DA-NER systems face additional challenges to the afore mentioned ones:

- Lack of annotated data for supervised NER;
- Lack of standard orthographies or language academics (Habash et al., 2013): unlike MSA, the same word in DA can be rewritten in so many forms, e.g. *mAtEyT\$*, *mtEyT\$*, *mA tEyT\$* ‘do not cry’ are all acceptable variants since there is no one standard;
- Lack of comprehensive enough Gazetteers: although MSA and DA named entities might overlap in cases such as location, if the author is following MSA writing. But, this is a problem facing all NER systems for all languages addressing NER in social media text, since by definition such media has a ubiquitous presence of highly productive names exemplified by the usage of nick names, hence the PER class will always face a coverage problem in social media; and
- applying NLP tools, such as NER, that are designed for MSA on DA results in considerably lower performance, thus the need to build resources and tools that specifically target DA (Habash et al., 2012).

Table 1: Morphological Features

Feature	Feature Values
Aspect	Verb aspect: Command, Imperfective, Perfective, Not applicable
Person	Person Information: 1st, 2nd, 3rd, Not applicable
Proclitic3	Question proclitic: No proclitic, Not applicable, Interrogative particle
Proclitic2	Conjunction proclitic: No proclitic, Not applicable, Conjunction <i>fa</i> , Connective particle <i>fa</i> , Response conditional <i>fa</i> , Subordinating conjunction <i>fa</i> , Conjunction <i>wa</i> , Particle <i>wa</i> , Subordinating conjunction <i>wa</i>
Proclitic1	Preposition proclitic: No proclitic, Not applicable, Interrogative <i>i\$</i> , Particle <i>bi</i> , Preposition <i>bi</i> , Progressive verb particle <i>bi</i> , Preposition <i>Ea</i> , Preposition <i>EalaY</i> , Preposition <i>fy</i> , Demonstrative <i>hA</i> , Future marker <i>Ha</i> , Preposition <i>ka</i> , Emphatic particle <i>la</i> , Preposition <i>la</i> , Preposition <i>li</i> + preposition <i>bi</i> , Emphatic <i>la</i> + future marker <i>Ha</i> , Response conditional <i>la</i> + future marker <i>Ha</i> , Jussive <i>li</i> , Preposition <i>li</i> , Preposition <i>min</i> , Future marker <i>sa</i> , Preposition <i>ta</i> , Particle <i>wa</i> , Preposition <i>wa</i> , Vocative <i>wA</i> , vocative <i>yA</i>
Proclitic	Article proclitic: No proclitic, Not applicable, Demonstrative particle <i>Aa</i> , Determiner, Determiner <i>Al</i> + negative particle <i>mA</i> , Negative particle <i>lA</i> , Negative particle <i>mA</i> , Negative particle <i>mA</i> , Particle <i>mA</i> , relative pronoun <i>mA</i>
Enclitics	Pronominals: No enclitic, Not applicable, 1st person plural/singular, 2nd person dual/plural, 2nd person feminine plural/singular, 2nd person masculine plural/singular, 3rd person dual/plural, 3rd person feminine plural/singular, 3rd person masculine plural/singular, Vocative particle, Negative particle <i>lA</i> , Interrogative pronoun <i>mA</i> , Interrogative pronoun <i>man</i> , Relative pronoun <i>man</i> , <i>ma</i> , <i>mA</i> , Subordinating conjunction <i>ma</i> , <i>mA</i> .

Approach For our proposed DA-NER system (Zirikly and Diab, 2014b, 2015), we use CRF sequence labeling as described in (Lafferty et al., 2001). CRF was chosen based on its superior performance over other machine learning approaches for NER task (Finkel et al., 2005). Additionally, (Benajiba and Rosso, 2008) demonstrates that CRF yields better results for Arabic NER.

Baseline For our baseline system (BL-NER), we reimplemented the features proposed in (Abdul-Hamid and Darwish, 2010) that produces near state-of-the-art results: previous and next word, and leading and trailing $\pm(2-4)$ character n-grams. These features have the advantage of being: directly applicable to DA as none of the features rely on the availability of morphological or syntactic analyzers, and they do not use any external gazetteers (e.g. Wikipedia gazetteers), as the gazetteers coverage in DA is more limited when compared to MSA, especially for *PER* class.

NER Features We introduce the following features in our NER system:

- **Lexical Features:** character n-gram features, the head and trailing bigrams (L2), trigrams (L3), and 4-grams (L4) characters;
- **Contextual Features (CTX):** The surrounding undiacritized words of a context window = $\pm 1(W_{-1}, W_0, W_1)$;
- **Gazetteers (GAZ):** We use the union of the following gazetteers: i) ANERGaz (Benajiba and Rosso, 2008), which contains 2183 locations, 402 organizations, and 2308 persons; and ii) WikiGaz: large Wikipedia gazetteer (Darwish and Gao, 2014), which contains 50141 locations, 17092 organizations, and 65557 persons. The entity mention and gazetteers’ entries matching methods used are the following:
 - Exact match (EM-GAZ): For more efficient search, we use the Aho-Corasick algorithm (Aho and Corasick, 1975) that has linear running time in terms of the input length plus the number of matching entries in a gazetteer;
 - Partial match(PM-GAZ): This feature is created to handle the case of compound gazetteers’ entries. If the token is part of the compound name then this feature is set to true. For example, if the gazetteer entry is *yAsr ErfAt* ‘Yasser Arafat’ and the input text is *yAsr BarakAt* then PM-GAZ for

the token *yAsr* will be activated. This is particularly useful in persons names as it recovers the list of first names in compounds; and

- Levenshtein match (LVM-GAZ): We use Levenshtein distance (Levenshtein, 1966) to compare the similarity between the input and a gazetteer entry. This is based on the observation that social media data contains non-standard spelling of words, due to the use of Dialectal Arabic.
- **Morphological Features:** The morphological features that we employ in our feature set (as shown in Table 1), are generated by MADAMIRA (Pasha et al., 2014) tool. The set contains:
 - Part of Speech (POS) tags within a window of ± 1 (POS_{-1} , POS_0 , POS_1);
 - Capitalization (CAPS): we check the capitalization of the translated NE that can represent a cue if the current token is NE or no (Benajiba et al., 2008). This feature is generated using the token’s English gloss;
 - Aspect (ASP), person (PERS), proclitics0 (PROC0), proclitics1 (PROC1), proclitics2 (PROC2), proclitics3 (PROC3), enclitics0 (ENC0); detailed description of these features is provided in Table 1;
 - isNum: Binary feature that is enabled if the token is a number; and
 - isNoun: Binary feature that is activated if the token is a proper noun (POS=noun prop).
- **Brown Clustering (BC):** Brown clustering (Brown et al., 1992) is a hierarchical clustering approach that maximizes the mutual information of word bigrams. It has been shown that word representations, especially Brown Clustering, improve NER systems’ performance (Turian et al., 2010). For each token, BC generates a cluster ID in the form of 1s and 0s. However, the length of the cluster ID differs between tokens.

In this work, for each BC cluster ID, we add variable prefixes length (4,5,6,7,10,13, and the full length of the cluster ID) as features resulting in: BC4, BC5, BC6, BC7, BC10, BC13, and BC features respectively. This group of features is based on the observation that semantically similar words will be grouped together in the same cluster; and

- **Word2vec:** Word2vec (Mikolov et al., 2013b) is a neural language model to learn word embeddings. Embeddings are represented by a set of latent variables, where each word is a specific instantiation of these variables, and finally presented as a vector of d dimensions. Word2vec uses either: a) Continuous Bag of Words (CBOW) model which predicts the word given its context; or b) skip-grams model which predicts the context given a word. Word embeddings have also been widely used in many NLP tasks and in NER (Turian et al., 2010).

Datasets We use two datasets:

1. Twitter dataset annotated using the Linguistics Data Consortium (LDC) tagging guidelines:
 - Training: Twitter dataset proposed in (Darwish and Gao, 2014), which contains 3,646 tweets (scraped using the query “lang:ar”) that are randomly selected from tweets that were authored in the period of May 3-12, 2012; and
 - Testing (Darwish, 2013): 1,423 tweets that were randomly selected from tweets authored between November 23, 2011 and November 27, 2011, and it has been used for testing in (Darwish and Gao, 2014).
2. Dialectal Arabic dataset (DA-EGY): The annotated data was chosen from a set of web blogs that are manually identified by LDC as Egyptian dialect and contains nearly 24k tokens. The data was annotated by one native Arabic speaker annotator who followed the LDC guidelines for tagging. We use the same 80/20 train/test 5-fold cross validation split proposed in (Zirikly and Diab, 2014a); and

3. Brown clustering and word2vec datasets. We generate three batches of output for each of the two using the following datasets: i) Newswire (NW) data of Arabic Gigaword, ANERCorp, and NW ACE2005 and ACE2006; ii) Broadcast News (BN) data of ACE2005 and ACE2006; iii) Weblogs (WL): Twitter data (training and testing), WL data of ACE2005 and ACE2006, and Arabic Dialect¹. The number of clusters k used for BC and word2vec is empirically chosen, where the best results achieved when $k=500, 200, 200$ for brown clustering on NW, BN, and WL respectively, as opposed to: 300,150,150 for word2vec on NW, BN, and WL respectively.

Table 2 shows dataset statistics for (1&2), namely number of tokens, and the named entity types: PER, LOC, and ORG.

Table 2: Twitter and DA-EGY dataset statistics

	#Tokens	#PER	#LOC	#ORG
Twitter-Train	55k	788	713	449
Twitter-Test	26k	464	587	316
DA-EGY	24k	311	155	19

Tools The Tools used are:

1. MADAMIRA (Pasha et al., 2014): tokenization and morphological features such as gender and POS tags; and
2. CRFSuite implementation (Okazaki, 2007).

3.2.2 Results and Discussion

Results We will use the abbreviated feature set as described in Table 3.

We use the weighted macro-average across the three NEs classes (PER, LOC, ORG) to calculate the overall performance in Table 4. Our proposed feature set generates competitive performance yielding an $F1=57.84\%$ without the use of any external resources vs. 59.59% with the use of gazetteers². Although word2vec and BC increase F1 about 10% over BL, we note that BC improvement (+6%) is more significant than word2vec with only 3% improvement. This is aligned with (Turian et al., 2010) observations that Brown Clustering yields the best English NER performance when compared to other word embedding approaches. The results for Levenshtein Matching approach (+ 8%) support our hypothesis that LVM-GAZ overcome the non-standardization issue in DA, when compared to EM-GAZ and PM-GAZ.

The results achieved are promising, especially in the area of social media since generating gazetteers, with high coverage, is a challenging and expensive task.

We study the impact of applying BC and word2vec on different data genre. We take as an example BC, shown in Figure 1. We note that genre variations impose minimum impact on word representations, which lead us to the conclusion that word2vec and BC presents robust and domain-independent features.

DA-EGY Results: Table 4 depicts the results of our NER system using the different proposed features and 5-fold cross validation. Our system outperforms the state-of-the-art results by about 7% with the use of gazetteers, and about 2% without the use of gazetteers. As shown in Table 5³, we notice that FS15, which uses word2vec features and excludes BC features and gazetteers, generates very comparable results (72.61%) to the best gazetteers-free performance achieved which achieved 72.68%.

¹LDC2012T09

²It should be noted that our use of the gazetteers is probably different from (Darwish and Gao, 2014), thus the difference in results.

³We omit ORG class in DA-EGY experiments because of its very limited occurrences (; 0.05% instances)

Table 3: DA-NER Feature Set

<i>Feature</i>	<i>Description</i>
Feature set1 (FS1)	Baseline (BL) features, as proposed in (Abdul-Hamid and Darwish, 2010), exact match against the Wikipedia gazetteers (WikiGaz)
Feature set2 (FS2)	BL features with the use of CAPS (English gloss capitalization) and the current, previous and next POS
Feature set3 (FS3)	FS2 + ENC0 + PROC0 + PROC1 + PROC2 + PROC3, as demonstrated in Table 1
Feature set4 (FS4)	FS3 + isNum + isNoun binary features
Feature set5 (FS5)	FS4 + word2vec cluster IDs
Feature set6 (FS6)	FS4 + BC cluster IDs with different prefixes length
Feature set7 (FS7)	FS6 + word2vec cluster IDs
Feature set8 (FS8)	FS7 + exact (EM-GAZ) + partial (PM-GAZ) match
Feature set9 (FS9)	FS8 + Levenshtein gazetteers' entires match with distance threshold set to 1 (LVM-GAZ1)
Feature set10 (FS10)	FS7 + Levenshtein gazetteers' entires match with distance threshold set to 2 (LVM-GAZ2)
Feature set11 (FS11)	FS8 + LVM-GAZ1 + LVM-GAZ2
Feature set12 (FS12)	FS11 + ASP + PERS morphological features
Feature set13 (FS13)	FS7 + ASP + PERS features
Feature set14 (FS14)	FS6 + ASP + PERS features
Feature set15 (FS15)	FS5 + ASP + PERS features

Table 4: Twitter DA-NER Results

	LOC	ORG	PER	Overall		
	F1	F1	F1	PREC	REC	F1
BL	49.15	38.38	48.78	83.25	31.77	45.99
BL+WikiGaz (Darwish and Gao, 2014)	65.5	41.5	48.5	79.3	42.1	55
FS1=BL + EM-GAZ (LOC,ORG,PER)	52.37	41.72	57.09	83.07	37.44	51.62
FS2=BL+CAPS+POS[-1,0,1]	51.23	38.6	59.2	79.89	38.1	51.59
FS3=FS2 +PROC{3,2,1,0}+ENC0	51.73	39.55	60.78	79.01	39.53	52.7
FS4=FS3+isNum+isNoun	51.56	39.48	60.91	79.11	39.53	52.72
FS5=FS4+word2vec	53.18	38.37	62.74	79.54	40.57	53.74
FS6=FS4+BC	55.41	40.46	65.02	81.83	42.6	56.03
FS7=FS6+word2vec	56.78	39.74	65.58	82.01	43.12	56.52
FS8=FS7+EM-GAZ+PM-GAZ	58.02	41.87	67.09	81.83	44.94	58.02
FS9=FS8+LVM-GAZ1	58.1	40.71	67.33	81.25	44.94	57.87
FS10=FS7+LVM-GAZ2	59.63	41.4	67.39	81.71	45.47	58.42
FS11=FS9+LVM-GAZ2	59.63	41.28	68.17	81.93	45.86	58.8
FS12=FS11+ASP+PERS	61.03	41.28	68.92	81.7	46.9	59.59
FS13=FS7+ASP+PERS	58.29	38.11	68.32	80.89	45.01	57.84

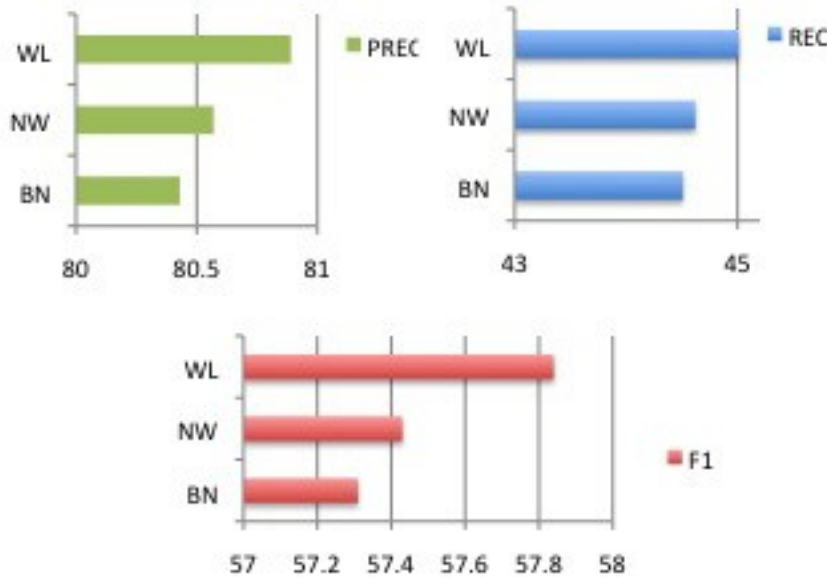


Figure 1: BC Data Genre and Performance correlation

3.3 Cross-Lingual Transfer Learning NER

3.3.1 Introduction

Previously proposed cross-lingual NER approaches use annotated tags projection via word-aligned bilingual parallel text (bitext) and used them to train sequential labeling models in the low resource targeted

Table 5: DA-EGY NER Results

	LOC	PER	Overall		
	F1	F1	PREC	REC	F1
State-of-the-art	91.43	49.18	86.53	62.3	70.31
FS12	96.77	57.47	82.9	72.39	77.12
FS13	89.66	55.7	86.67	63.08	72.68
FS14	89.66	54.05	90	61.04	71.86
FS15	89.66	55.56	93.48	61.04	72.61

languages (Yarowsky et al., 2001; Das and Petrov, 2011), and (Wang and Manning, 2014)). However, this faces two main issues:

- Generate sub-optimal models due to the noisiness generated by projecting the tags; and
- Limited availability of bitext especially for resource-poor languages and domains, where it is often that low resource languages are moderately-sized monolingual/comparable corpora and have small bilingual dictionaries

Thus, we experiment with a *direct transfer* approach on cross-lingual NER, also classified as transductive transfer learning (Pan and Yang, 2010).

We test our proposed approach, depicted in Figure 2, on the e-commerce dataset that is characterized by being noisy and not very structured. We build a model that is trained on English data (high resource language) and transfer the model to Spanish (ES) and Chinese (ZH).

The results observed, albeit the proposed constraints (e.g. significantly small training data), are sufficiently motivating to generalize the same system framework to DA-NER.

Contribution: To our knowledge, we are the first who apply the direct transfer learning to NER without the use of parallel data.

3.3.2 Methods, Assumptions, and Procedures

Approach Our transfer learning approach relies on a mixture of mono and cross lingual transferable features.

Monolingual NER in Source Language

Our framework is based on the *direct transfer* approach, where we extract abstract, language-independent and non-lexical features FS and FT in LS and LT. A subset of FT is generated using a mapping scheme discussed in Section 3.3.2, then, directly apply LS NER model on LT using FT. We adopt Conditional Random Field (CRF) sequence labeling as described in (Lafferty et al., 2001) to generate the English model.

Most NER systems use lexical features (e.g. n-gram characters, previous or next words), however, due to the cross lingual nature of our direct transfer approach, we cannot rely on any lexical features as they are not transferable. Thus, we define a set of features that are language independent (Zirikly and Hagiwara, 2015):

Monolingual Features

1. *Token position*: instead of using the token exact position, we use token relative position in addition to position’s binary features such as token is in: first, second, and last third of the sentence. These features are based on the observation that certain tokens, such as brand names in title or description of a product, tend to appear at the beginning of the sentence, while others toward the end;
2. *Word Shape*. We use a list of binary features: is-alphanumeric, is-number, is-alpha, is-punctuation, length of number (if is-num is true), pattern-based features (e.g. regular expressions to capture certain patterns such as products model numbers), latin-only features (first-is-capital, all-capitals, all-small);

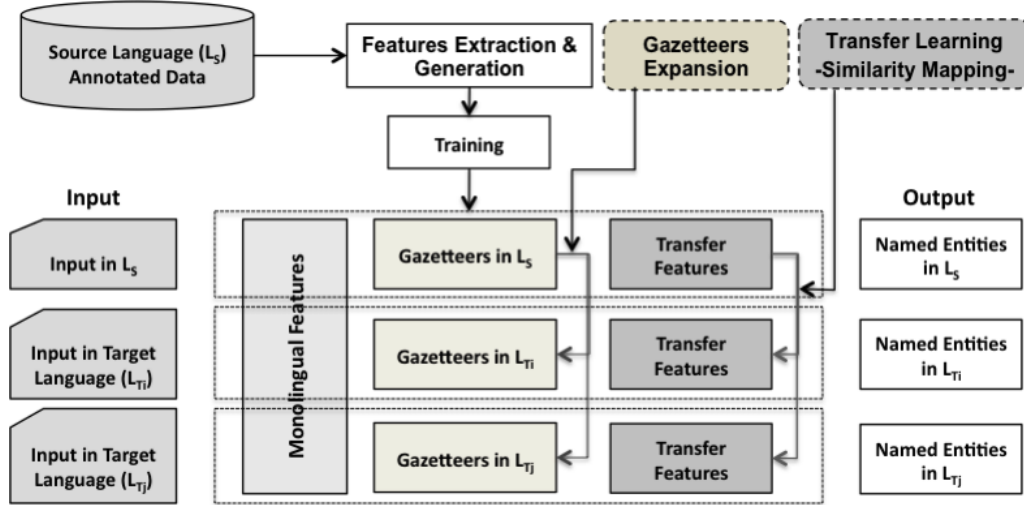


Figure 2: Transfer Learning System Framework for Low resources Languages

3. *In-Title*: a binary feature that specifies whether the token is in the product’s title or description. For instance, brand names mostly appear in the beginning of titles, while this does not hold in descriptions;
4. *Preceding/Proceeding keywords within window*: some NEs are often preceded by certain keywords. For instance, often a product size is preceded by certain keywords such as dimension, height or the word “size”. In our work we use a manually created list of keywords for two classes, Color and Size. Although the keyword list is domain dependent, it is often short and can be easily updated;
5. *Universal Part of Speech Tags*: Part of Speech (POS) tags have been widely used in many NER systems. However, each language has its own POS tagset that often has limited overlap with other POS languages’ tagsets. Thus, we use a coarse-grained layer of POS tags called Universal POS, as proposed in (Petrov et al., 2011);
6. *Token is a unit*: a binary feature that is set to true if it matches an entry in the units dictionary (e.g., “cm.”);
7. *Gazetteers*: building dictionaries for every LT of interest is expensive; thus, we propose a method, described in Section 3.3.2, to generate gazetteers in LT given LS gazetteers; and
8. *Brown Clustering (BC)*: word representations, especially Brown Clustering (Brown et al., 1992), are used in many NLP tasks and are proven to improve NER performance (Turian et al., 2010). In this work, we use cluster IDs of variable prefix lengths in order to retrieve word similarities on different granularity levels.

Multilingual NER in Target Language Our goal is to transfer each feature from LS to LT space. The main challenge resides in transferring features 7 & 8 without the use of external resources and parallel data for every targeted language.

Brown Clustering Mapping Given i) Vocabulary in the source/target languages, and ii) The output of brown clustering on LS and LT, we aim to find the best mapping c^{S*} that maximizes the cluster similarity $simc$ for each target cluster (Equation 1), and for each metric discussed in the following paragraph. We calculate the cluster similarity $simc$ as the weighted average of the word similarity $simw$ of the two clusters members (Equation 2).

$$c^{S*} = \arg \max_{c^S \text{ in } C^S} CSSimC(c^S, c^T) \text{ for each } c^T \text{ in } C^T \quad (1)$$

$$simc(c^T, c^S) = (1/|c^S||c^T|) \sum_{w^S \text{ in } c^S, w^T \text{ in } c^T} simw(w^S, w^T) \quad (2)$$

Clusters Similarity Metrics The similarity metrics used are as follows:

- *String Similarity* (external resources independent): This metric works only on languages that share the same alphabet, as it is based on the intuition that most NEs conserve the name’s shape or present minor changes that can be identified using edit distance in closely related languages (we use Levenshtein distance (Levenshtein, 1966)). The two variations of string similarity metrics used are: i) *Exact match*: $simw(w_i, w_j) = 1$ if $w_i = w_j$; and ii) *Edit distance*: $simw(w_i, w_j) = 1$ if $levenshtein\text{-}distance(w_i, w_j) < \theta$.
- *Dictionary-based similarity*: We present two similarity metrics using BabelNet synsets (Navigli and Ponzetto, 2012). First, we use *Binary co-occurrence* $simw^{binary}(w_i, w_j) = 1$ if w_j in $synset(w_i)$, where $synset(w_i)$ is the set of words in the BabelNet synset of w_i . Second, we use *Frequency weighted*. This is a weighted version of the binary similarity that is based on the observation that less frequent words tend to be less reliable in brown clustering: $simw^{weighted}(w_i, w_j) = [\log f(w_i) + \log f(w_j)] \times simw^{binary}(w_i, w_j)$ where $f(w)$ is the frequency of word w . Unlike String similarity metrics, these metrics are not limited to similar languages due to the use of the multilingual dictionary BabelNet, which covers 271 languages.

Gazetteer Expansion We use graph-based semi-supervised learning to expand the gazetteers from LS to LT using an intra-language similarity model. This indirect inference of beliefs from the source gazetteers to the target can be modeled by semi-supervised graph propagation (Zhu et al., 2003), where graph nodes are $V_S \cup V_T$, positive labels are entries in the LS gazetteer (e.g., GPE) which we wish to expand to LT, and negative labels are entries in other gazetteers (e.g., PERSON) in LS. The edge weights between same-language nodes w_i and w_j are given by $\exp(-\|w_i - w_j\|)$ where w_i is the distributed vector representation of word w_i computed by word2vec (Mikolov et al., 2013b). The edge weights between node w_i in V_S and w_j in V_T are defined 1 if the spelling of these two words are identical and 0 otherwise. Note that this spelling based similarity propagation is still available for language pairs with different writing systems such as English and Chinese, because major NEs (e.g., brand names) are often written in Roman alphabets even in Chinese products. Since the analytical solution to this propagation involves the computation of $n \times n$ (n is the number of unlabeled nodes) matrix, we approximated it by running three propagation steps iteratively, namely, $L_S \rightarrow L_S$, $L_S \rightarrow L_T$, and $L_T \rightarrow L_T$. After the propagation, we used all the nodes with their propagated values $f(w_i) \gg \theta$ as entities in the new gazetteer.

Baseline We choose a strong Machine Translation baseline, since to the best of our knowledge, there is no previous work that proposes transfer learning for NER without the use of parallel data. Given the language pair (L_S, L_T) , we use Microsoft Bing Translate API to generate $L_S \rightarrow L_T$ translation. Then, we apply LS NER model on the translated text and evaluate by mapping the tagged tokens back to LT using the word alignments generated by Bing (free as opposed to Google Translate).

Table 6: Transfer Learning Dataset Language-Tags Numbers Stats

	Color	Brand	Material	Model	Type	Size
EN	358	814	733	203	1238	427
ES	207	425	301	172	606	126
ZH	416	60	381	24	690	306

Datasets We collect the list of products (titles and descriptions). The titles of products are about 10 words long and poorly structured, adding more difficulties to our task. On the other hand, the length of product descriptions ranges from 12-130 words. The e-commerce domain poses the need to introduce new NE tagset as opposed to the conventional ones; we introduce 6 tag types: 1) Color; 2) Brand names; 3) Size; 4) Type: e.g. “camera”, “shirt”; 5) Material: e.g. “plastic”, “cotton”; 6) Model: the model number of a product: e.g., “A1533.”. For the rest of the experiments, the source language is going to be English (EN), whereas the target languages are Spanish (ES) and Chinese (ZH). The datasets used in our experiment’s settings are: i) Training data: 1800 annotated English products from Rakuten.com shopping (Rakuten, 2013a); ii) Test data: 300 ES products from Rakuten Spain (Rakuten, 2013b) and 500 products from Rakuten Taiwan (Rakuten, 2013c); iii) Brown clustering: *English*: Rakuten shopping 2013 dump (19m unique products with 607m tokens); *Spanish*: Rakuten Spain 2013 dump (700K unique products that contains 41m tokens) in addition to Spanish Wikipedia dump (Al-Rfou’, 2013); *Chinese*: Wikipedia Chinese 2014 dump (147m tokens) plus 16k products crawled from Rakuten Taiwan. Table 6 shows the numbers of tags per category for each language.

3.3.3 Results and Discussion

Results For each studied language we use Stanford CoreNLP (Manning et al., 2014b) for EN and ZH, and TreeTagger (Schmid, 1994) for ES to produce the tokens and the POS tags. However, we apply extra processing steps to the tokenizer due to the nature of the domain’s data (e.g., avoid tokenization in some instances that involve punctuations), in addition to normalizing URLs, numbers, and elongation. We also map POS tags for all the source and target languages to the universal POS tagset as explained in 3.3.2.

We apply the conventional (80:20 train/test split) and report our results based on 5-fold cross validation throughout all the transfer learning experiments. Based on Table 7, we note that English monolingual performance is considerably lower than state-of-the-art English NER systems, which is expectable due to: the targeted domain’s nature, the newly proposed NE tagset, and most importantly, the considerably small training data (1280 products). These factors inherently affect the baseline and our proposed system performance.

Table 7 illustrates the results for the English monolingual NER system (EN-Mono), baseline for ES and ZH (ES-Baseline and ZH-Baseline, respectively) and our proposed transfer learning approach with the gazetteer expansion (ES-TL and ZH-TL). Additionally, we add the results of our proposed approach where the gazetteers used are machine translated using Google translate from the English gazetteers to Spanish (ES-MT), in order to evaluate our gazetteer expansion approach performance to the translated gazetteers.

The poor baseline results are due to the poor word alignment (i.e. incorrect tag projection) generated by Bing Translate, which is mainly due to the noisy nature of the domain’s data, which we believe that its effect would be less in Newswire data, but not necessarily in weblogs.

3.4 Transfer Learning MSA-DA NER

3.4.1 Introduction

As we have mentioned earlier, one of the main challenges that DA-NER faces (similarly to other low resource languages) is the lack of annotated data. Thus, approaches that learn from high-resource languages’ annotated data (in our case MSA) to predict the label of low resource targeted language (e.g. DA) are highly

Table 7: F-score Results

	Color	Brand	Material	Model	Type	Size	Micro-Avg
EN-Mono	68.45	71.91	50.94	59.78	53.73	45.42	61.12
ES-Baseline	24.23	3.44	13.08	14.51	12.5	6.61	13.79
ES-TL	18.00	9.37	8.05	16.99	18.26	10.64	39.46
ES-GT	38.49	13.31	33.5	2.27	36.43	1.16	30.20
ZH-Baseline	19.16	2.79	11.96	None	9.35	6.34	12.58
ZH-TL	9.36	1.02	1.81	None	17.28	17.74	23.43

desirable. Based on this observation, we implemented our cross-lingual transfer learning (TL) approach described in the previous section 3.3 for MSA/DA setting. Although, we previously proposed gazetteers expansion, but in this setting, we opt not to expand our MSA gazetteers. We reason that with the fact that DA and MSA are two variants of the same language group, thus the need of gazetteers expansion might be of less necessity. Additionally, the expansion process in this setting needs to be updated to handle not only the different orthographic ways of writing the same named entities (especially when it comes to locations), but rather to handle the nicknames used for *person* named entities, which is commonly used in social media. Furthermore, because of the special language pairs (MSA and DA), we can take advantage of some of the language-dependent features (e.g. character n-grams).

In this section, we will briefly give a reminder of the features used in the monolingual MSA-NER system proposed in Section 3.2 using CRF and mark the features used in our TL approach. Next, we will describe the datasets used and present the results.

3.4.2 Methods, Assumptions, and Procedures

MSA-NER *Monolingual* features The monolingual features used are adopted from the features in Section 3.2 with the same naming, where we keep language-dependent features due to the special nature of this setting where cross-lingual in this context is actually ~~cross~~-cross-dialects. However, we think it is worth reporting the results with/without language-dependent features shown in Section 3.4.2, and draw some conclusions on the importance of lexical features in such models.

- Lexical features (LEX): character n-grams (L2, L3, L4);
- Contextual features (CTX): the surrounding words in the window of $[-1, +1]$;
- Gazetteers co-occurrence features based on exact and partial match, in addition to Levenshtein distance= 2;
- Morphological features: we opt not to use this feature due to its dependency on tools that can process multi-dialectal text, additionally we believe that these features are less, or in some cases never, transferable for future work. Though MADAMIRA (Pasha et al., 2014) handles the Egyptian dialect, our dataset contains a mix of dialects. This makes any approach less appealing that relies on specific features such as CAPS (words that start with capital letters) of the English gloss, since we will have cases in which the gloss is not available. From the proposed morphological features introduced in Section 3.2, we use the language/dialect independent features namely isNum and add to it isPunc; both binary features that are enabled if the token is a number or punctuation, respectively; and
- Brown clusters (BC): we use brown clusters as features in the source language. After generating brown clusters for both MSA and DA, we apply our projection mechanism, as described in the previous chapter (Section 3.3.2), in the direction from DA_{BC} $\rightarrow$ MSA_{BC}.

Experiments

Datasets For building the high-resource (e.g. MSA) model, we use the following MSA annotated data for NER task: ANERCorp, ACE2005, ACE2006, and AQMAR. For testing our approach on dialectal data, we use the same test Twitter dataset (Darwish and Gao, 2014). For building the brown clusters for both MSA and DA, we use the following datasets: Arabic Gigaword, Newswire (NW) data of ACE2005 and ACE2006; Broadcast News (BN) data of ACE2005 and ACE2006; and ANERCorp. For DA brown clusters we use: Twitter data (training) and WL data of ACE2005 and ACE2006. The number of clusters k used for BC is empirically chosen with $k = 200$.

3.4.3 Results and Discussion

We note that our proposed TL approach yields an overall F1-score of 61.3 that surpasses the performance of our DA-NER system results of 59.6 depicted in Table 4. Based on the results shown in Table 8, character n-gram features improve the performance across dialects. We reason this behavior that dialectal and MSA words have a higher overlap in letters which n-grams will be able to capture. However, on the other side, we note that CTX features hurt the performance the most, which is a result of being the most *dialect*-dependent features where the overlap between MSA and dialectal words/tokens will be minimal, since DA doesn't necessarily conform to MSA grammatical rules of structuring sentences, especially with the number of characters limitation.

We believe that our work is the first that uses clustering projection and a transfer learning approach for the Arabic NER task. The results attained are evidence of not only the positive role that entity clusters play in NER tasks, as we proved in our previous work (Zirikly and Hagiwara, 2015) and (Zirikly and Diab, 2015), but additionally an indicator that TL approaches for MSA-DA knowledge transfer are efficient, especially that both *dialects* belong to the same family group, *Arabic*. Based on the results in Table 8, we note that the precision performance in the transferable models are comparable to the ones in MSA, however the degradation in performance is a result of low recall. Additionally, we note that PER class generates the worst results per entity type and this is a result of the frequent use of nicknames (e.g. Hamwdeh) in spoken and dialectal data (Zirikly and Diab, 2015). This is more evident in the performance score variance of PER entity type between MSA and DA unlike ORG and LOC where the variance is much smaller. It is also worth noting that the reported results are generated using the CoNLL script using IOB tagging, where the model is penalized for not identifying the correct named entity order (e.g. identifying I-PER as B-PER). We note that ORG named entity type poor performance is consistent between MSA and DA and with the results in Table 4.

Table 8: MSA & MSA to DA TL performance results

		LOC	ORG	PER	overall		
Features	dialect	F1	F1	F1	PREC	REC	F1
all-LEX-CTX	MSA	72.2	45.2	80.3	85.8	59.2	70.1
	DA	61.9	42.1	69.2	83.6	46.9	60.1
all-CTX	MSA	76.6	46.8	82.7	88.2	61.7	72.6
	DA	62.3	42.6	69.8	84.9	48	61.3
all-LEX	MSA	72.8	45.1	80.6	86.1	60.2	70.8
	DA	61.6	41.8	68.7	83.6	46.5	59.8

3.5 Deep Neural Networks DA-NER Model using Multi-representation Embeddings

3.5.1 Introduction

Neural Networks in general have proven to deliver promising results in many NLP tasks (Collobert and Weston, 2008). Recent studies and systems have shown that Bidirectional Long-Short Term Memory (Bi-LSTM) are specifically useful for sequence labeling tasks (e.g. POS and NER tagging), especially when a CRF layer is added on the top of the NN architecture Lample et al., 2016b; Huang et al., 2015. However, the previous work in this area has proven to be especially beneficial for high-resource languages (e.g. English) where large training data is available (Huang et al., 2015). In this section, we will first describe the structure of the Bi-LSTM-CRF we used, depicted in Figure 3, then we will cover the input features employed. We will also introduce new types of embeddings as input for the neural network, that would replace the conventional word embeddings presented in (Mikolov et al., 2013b) that the majority of neural network models in NLP use with a more abstract representation that requires no training (unlike word embeddings), and is dependent on our previous work on lemma identification.

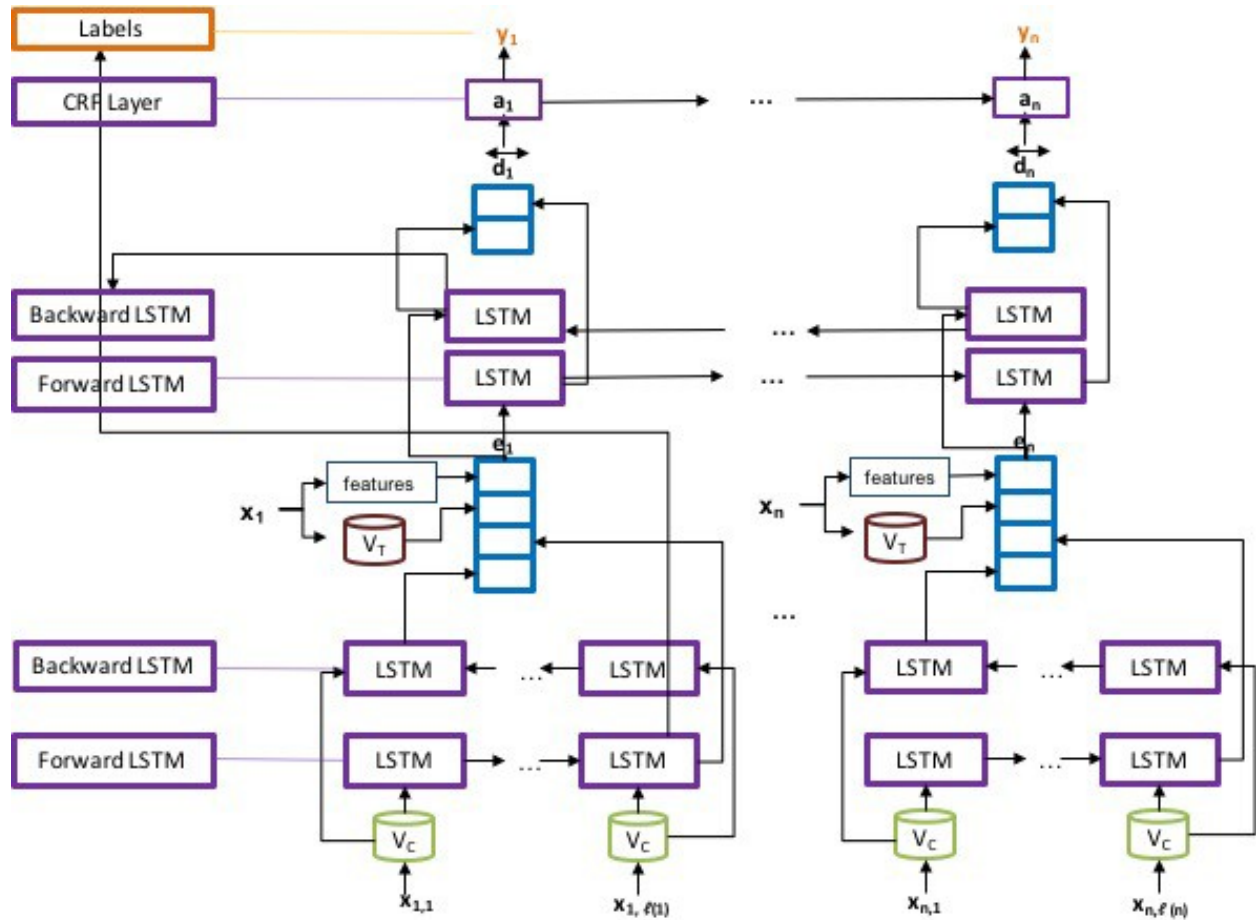


Figure 3: Bi-LSTM-CRF DA-NER Model

3.5.2 Methods, Assumptions, and Procedures

LSTM networks have the advantage of incorporating long distance features into the prediction, hence they are memory-enabled networks. Although both Recurrent Neural Networks (RNN) and LSTMs are memory-based models, LSTMs have the advantage of handling long-term dependency problems because LSTMs

are built to avoid exploding/vanishing gradient problems, where the gradient becomes very large or small in the earlier layers of the network, thus effecting the learning rate. LSTMs avoid these two problems by using the identity function as the activation function with a derivative of 1.0, thus the back-propagated gradient neither vanishes nor explodes when traversing the layers, but remains constant. Figure 4 shows the details of an LSTM cell, depicting the mechanism of avoiding exploding and vanishing gradients.

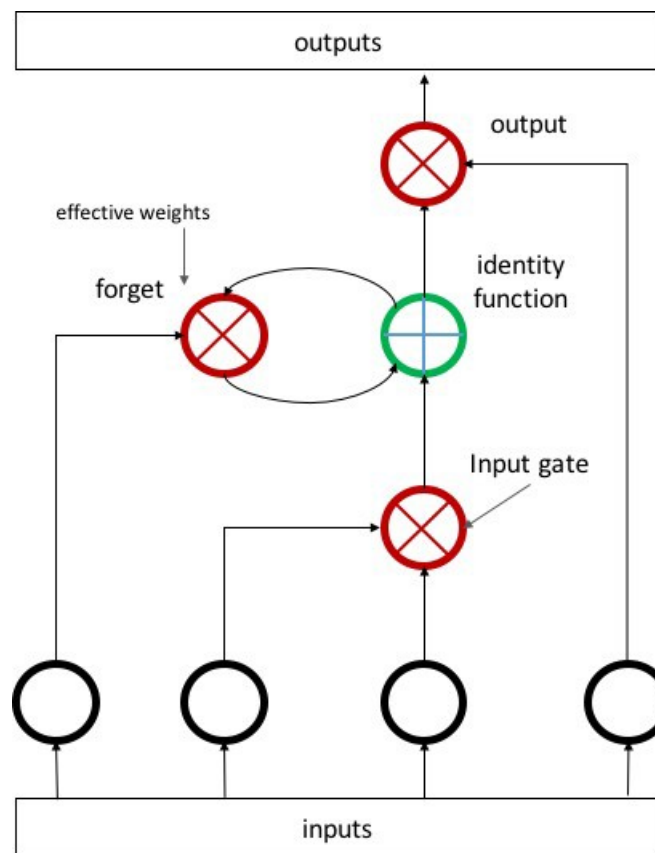


Figure 4: LSTM Cell

(Dernoncourt et al., 2017) have shown that character and token LSTMs produce almost state-of-the-art results for the English NER task on standard datasets in the field: CoNLL: 90.5 vs 90.9 state-of-the-art, and i2b2-2014: 97.7 vs 97.9 (best published results on patients de-identification task). However, it is worth noting that the best results (Passos et al., 2014) for CoNLL (the standard dataset for English NER) are achieved using a CRF model, however the authors incorporate the use of phrase embeddings that are based on lexicons which results in boosting the performance.

Our DA-NER system uses the following structure and input:

1. A character-based Bi-LSTM that produces character embeddings V_c based on the input token x_i , where L_i represents the number of characters, and i is the index of the current token with $i \in [1..n]$: n is the number of tokens in the corpora.
2. A token-based Bi-LSTM that takes as input token embeddings such as word vectors (Mikolov et al., 2013b) or any other kind of embeddings.
3. Gazetteer features: identical to what is described in 3.2.1.

Table 9: AQMAR dataset stats

#Tokens	PER	LOC	ORG
74000	2405	2048	960

4. Word shape features: binary features that are enabled if the token is a number (isNum), or a punctuation (isPunc).
5. The resulting vectors are concatenated into a vector that is fed to another bi-directional LSTM, resulting in a vector which is the input for the CRF layer that outputs the probabilities and the final labels predictions.

Token Embeddings We experiment with multiple types of embeddings starting from the commonly used embeddings using word2vec (Mikolov et al., 2013b), and more recent embeddings fasttext (Bojanowski et al., 2016) that is trained on Wikipedia data, our own proposed patterns with augmented roots lexicon (Habash and Rambow, 2006).

Pattern-based Embeddings Almost all NLP systems that use NN models depend on character and token embeddings or the combination of the two, where the latter is dependent on pre-trained models generated by neural network models that are trained to produce word embeddings (e.g. word2vec or a variation of it). In our NER system, we propose a new embedding that aims to take advantage of the special characteristics of the Arabic language. Specifically, we exploit the root-and-pattern nature of Arabic morphology, which helped us identify the lemmas. In our work, we replace the conventional embeddings that require large amounts of training data with our generated patterns that don’t require pre-training. Additionally, due to the large variation of Arabic dialects, token embeddings need to be generated per language dialect; in contrast, our proposed approach is dialect-agnostic and depends only on the idea that named entities, regardless of the dialect, are mostly mapped to a finite closed set of patterns, especially for *person* and *location* entities. For both Arabic pattern databases we use our proposed database (Attia et al., 2016) augmented with (Habash and Rambow, 2006). We follow the same process after cleaning and tokenizing the text.

Experiments

Datasets As we are focusing on the DA-NER task, we use the same twitter dataset that we used above and proposed in (Darwish and Gao, 2014). We augment the training dataset with the MSA annotated corpus for the NER task where we use:

- The AQMAR Arabic Wikipedia Named Entity Corpus (Schneider et al., 2012), that is based on 28 Wikipedia articles that are hand annotated. Table 9 depicts the exact number of named entities in this dataset. It is worth noting that this dataset contains other NE types in addition to *PER*, *LOC*, *ORG* types, such as different types of miscellaneous *MIS* tag (4 categories that cannot be categorized under the above three categories). Such examples from the dataset are *Proton* which is tagged as B-MIS1.
- The ANERCorp (Benajiba et al., 2007) corpus as used in the previous NER proposed system.

3.5.3 Results and Discussion

In our experiments we examine two optimization methods (Ruder, 2016), namely: Stochastic Gradient Descent (SGD), and Adam (Kingma and Ba, 2014). We report our results based on SGD only since it produces better performance. We reason that by the ability of SGD to generalize better than adaptive optimizers such as Adam, as proven in (Wilson et al., 2017), even if adaptive optimizers might produce better results during training.

In the results section we describe the experiments under multiple settings where we opt to use dialectal only training data and then we augment it with MSA, keeping the test data intact. In both settings we use the following input:

- Embeddings on the token or lemma level:
 - Token embeddings using Wikipedia-generated word vectors (Bojanowski et al., 2016)
 - Patterns embeddings using our proposed method
- Gazetteers partial (PM-GAZ), full match (EM-GAZ) and Levenshtein-based (LVM-GAZ) occurrence. This feature value is the same as described in 3.2.1
- Binary word shape features:
 - isPunc: This feature is enabled when the token is a punctuation; and
 - isNum: This feature is enabled when the token is a number.

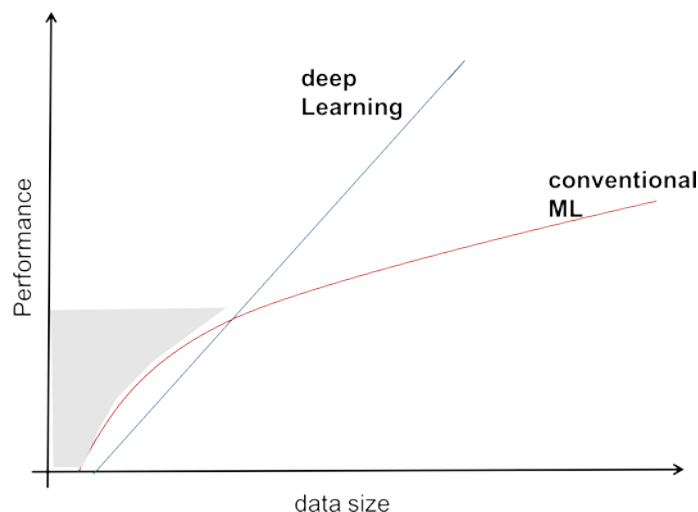


Figure 5: Size vs. Performance in deep learning and conventional ML; adopted from Andrew Ng slides on Neural Networks and Deep Learning

Table 10 shows the results of our proposed DA-NER system using the dialectal data where gazetteer features are both enabled and not. Additionally, this table shows the results when using word embeddings only. We note that using solely word embeddings we achieve almost the same results 57.3 as CRF models 57.84. Although deep learning has proven to generate *[almost]* state-of-the-art results for many NLP tasks, neural networks can provide best results when trained on large training data. As shown in figure 5, there is an area where conventional machine learning algorithms will outperform the results of neural network models, when training data size is small, which explains our results.

Table 10 shows the results using pattern (one-hot vectors) embeddings as opposed to word vector embeddings. We note that the results given our proposed pattern embeddings (56.9 vs. 58.76 without/with gazetteers, respectively) are comparable to word embeddings which require training on large data. These results validate our hypothesis that Arabic patterns, as an abstract representation of words, can play a significant impact on the role of the word in terms of NER (e.g. NER vs. not NER). This is the first trial of using Arabic patterns to recognize named entities. As a result we think that building pattern lexicons with high coverage and with the right preprocessing, even if it is applied on dialectal noisy data, can achieve promising

results that do not require the expensive word embedding training that is highly prone to the dialect, domain and genre variations of the data.

Table 10: DA-NER system performance results

type	precision	recall	F1	accuracy	gaz	pattern	
LOC	82.23	48.81	61.26		no	no	
ORG	68.64	36.48	51.61				
PER	67.51	51.61	58.5				
overall	61.26	58.5	57.3	95.56			
LOC	83.9	50.2	63.1		yes		no
ORG	70.1	38.3	53.2				
PER	69.7	53.7	60.1				
overall	63.5	60.3	59.1	96.8			
LOC	81.7	48.1	60.5		no	yes	
ORG	67.5	35.8	50.4				
PER	66.2	50.3	57.6				
overall	72.8	46.1	56.9	94.9			
LOC	83.4	49.6	62.1		yes		yes
ORG	69.8	37.1	52.2				
PER	68.8	52.7	59.3				
overall	74.5	47.6	58.76	96.1			

As we have previously explained at the beginning of Section 3.1.2, DA-Arabic NLP tasks pose multiple challenges, to list one of the most salient ones: the lack of training labeled data. This limitation comes from the large variety of Arabic dialects (at least 8 dialects⁴ in addition to the small dialectal nuances per country/city) and the inherited noise in the data coming from social media. Due to this limitation and the lack of an NER-annotated dialectal dataset, we opt to augment our training dataset with MSA using the following datasets: ANERCorp and AQMAR. We are aware of the noise that this addition might cause during the training phase, where this data imposes challenges on two folds: language (MSA vs. Dialectal) and different genre/domain. Nonetheless, the choice of augmentation was based on the intuition that the orthography of named entities is more consistent and less prone to dialects. This intuition is more evident when using our proposed pattern embeddings since the input, unlike word vectors, does not rely on the neighboring content (e.g. bigrams in word2vec). Table 11 shows the final results for both word vector embeddings and pattern embeddings. We note that pattern embeddings, despite the limited available pattern lexicon, generate comparable results (66.37 vs. 68.39, without/with gazetteers, respectively) to word vectors. Although we do not see significant improvements on the results when augmenting the training dataset, we opt to explain that MSA is largely different than dialects and the genre of the dataset (twitter vs. Wikipedia and newswire) represent an extra noise factor. However, as expected, the F1-score of identifying the LOC named entities has been mostly impacted given the consistency of locations across MSA and dialects, unlike other named entity types. For instance, in the PER entity type, nicknames (e.g. Abw mHmwd), as we previously mentioned, have stronger presence in dialects as opposed to MSA. Additionally, we note that our proposed model generates state-of-the-art results, when compared to the previous state-of-the-art results 65.2 (Darwish and Gao, 2014).

⁴<http://istizada.com/complete-list-of-arabic-speaking-countries-2014/>

Table 11: MSA-augmented DA-NER system performance results

type	precision	recall	micro-F1	accuracy	gaz	pattern	
LOC	84.5	71.7	77.6		no	no	
ORG	77.3	44.6	56.5				
PER	68	48.7	56.7				
overall	77.7	57.5	66.4	96.7			
LOC	85.8	72.6	78.6		yes		no
ORG	78.6	45.5	57.4				
PER	69.7	49.6	57.7				
overall	78.4	58.7	68.3	97.9			
LOC	83.9	71.3	77.1		no	yes	
ORG	76.8	44.2	55.9				
PER	67.6	48.2	56.3				
overall	77.3	56.9	65.61	96.2			
LOC	85.7	72.6	78.9		yes		yes
ORG	78.5	45.6	57.5				
PER	69.3	49.9	57.8				
overall	78.6	58.7	67.8	97.4			

3.6 Conclusions

In this chapter, we have discussed in some detail our work on named entity recognition. We now summarize our contributions in the area of NER:

- We have annotated a dataset for the NER task that is identified as Dialectal Data by the Linguistics Data Consortium (LDC), collected and annotated from weblogs with PER, LOC, and ORG entity types.
- We have built the DA-NER system that can handle weblogs and microblogs, and yields stable and high results **without** the use of gazetteers which makes it applicable to a wider set of genres, using word and entity clusters as an alternative to gazetteers.
- We have delivered a Java-based NER system for Modern Standard Arabic (MSA) that is integrated with MADAMIRA tools (Pasha et al., 2014).
- We have developed a transfer learning NER model targeting low-resource languages that is fully language-independent and transferable. The model is tested on Chinese and Spanish as “low-resource” languages from English. Additionally, we propose clustering projection and gazetteers expansion techniques that are expandable to other tasks that targets low-resource languages (e.g. urgency detection).
- We have developed and implemented a cross-lingual transfer NER model using transfer learning methodologies to generate NER models in low-resource languages (any language that lacks the availability of annotated data; examples from our scenario: Spanish and Chinese) from high resource language (e.g. English) in the e-commerce domain.
- We have applied our proposed transfer learning model to generate the DA-NER system for dialectal Arabic from high-resource settings (MSA).
- We have developed the first Dialectal-Arabic NER system that uses the transfer learning technique to transfer the knowledge from MSA to dialectal Arabic. The system is very similar to the one proposed

above with minor modifications that reflect the source and target languages settings: in this setting, we can include some language-dependent features, given the similarities between MSA and other dialects.

- We have developed a deep learning Dialectal-Arabic NER system that is based on Bi-LSTM with a CRF layer to generate NER labels. The system uses novel types of embeddings; namely pattern embeddings that are based on the patterns and templates as shown above, in addition to character embeddings. The main intuition for this system is that named entities of specific type in Arabic follow specific patterns. The system generates state-of-the-art results for DA-NER systems that target social media.

Our extensive work shows that there are many options for improving NER in situations in which we do not have extensive training data or auxiliary data in the target language.

Chapter 4

Entity Linking

4.1 Introduction

In this and the following section we will cover the last component of our framework that targets clustering, disambiguating and mapping named entities to entries in the Knowledge Base (KB). We will start with the supervised linking approaches (Entity Linking task) that involve correctly linking the name mentions in free text to entries in a KB. In the next chapter, we will discuss the unsupervised clustering aspect of the problem with a focus on true-aliases cases (no string overlapping between the entity's mentions). An example of entity linking is depicted in Figure 7, where we note that mapping Barack and Michelle Obama in the first sentence to the correct entries in the KB can be achieved with a high confidence score using exact string matching. However, linking *Obama* in the second sentence is not as intuitive and requires semantic knowledge in order to perform the right mapping.

4.2 Methods, Assumptions, and Procedures

Most of the linking techniques in the literature use supervised approaches that take as input the gold labeled name mentions with the correct KB identifier. For each name mention in the text, a list of all the candidates from the KB are generated, then a ranking method will be applied on the generated candidates list. Supervised and unsupervised approaches have been employed, with superior performance of supervised methods (Ji et al., 2014). Supervised approaches encompass binary classification, learning to rank, probabilistic, graph-based methods, in addition to recent approaches that introduce deep learning models (Francis-Landau et al., 2016) and entity embeddings (Ganea and Hofmann, 2017). Most state-of-the-art linking systems use Support Vector Machines (SVM) as their ranking approach (Guo et al., 2013; Hoffart et al., 2011).

Although our main focus concentrates on the alias instances that do not correspond to well-known entities, we still include a supervised entity linking system that can augment the knowledge representation in our aliasing system and help improving F1-score. We focus on English EL to evaluate the applicability of our approach.

We make the assumption that we have gold name mention (query) as input, where the input encompasses the following:

- Query mention string (e.g. Barack Obama, Obama); and
- The document where the mention string occurs in, along with the offset.

We adopt the above input format as proposed in TAC-KBP Entity Detection and Linking (EDL) task (Ji et al., 2015).

For each queried mention, we either provide the KB unique identifier (if we found an entry in the KB) or NIL. The latter case is what our proposed aliasing system *NEAR* is targeting, where we apply clustering

to group mentions that represent the same entity together. Figure 6 shows our proposed resolution system process.

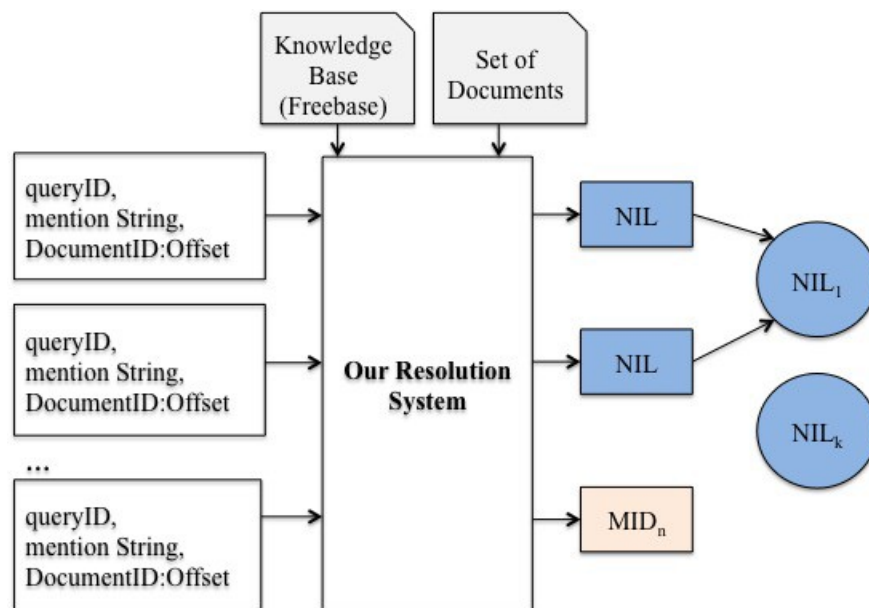


Figure 6: Resolution System

Preprocessing

There are several preprocessing steps that need to be performed on the mention string (e.g. name normalization), the documents, and some textual information related to the query in the KB. The list of preprocessing in more details includes:

1. **Name Normalization.**: examples of name normalization are: “BarackObama” becomes “Barack Obama”, and “OBAMA” becomes “Obama”;
2. **Preprocessing on Documents.** For the source text files, we perform: Tokenization (TOK), Part of Speech (POS) tagging, NER (including entity types), and Coreference Resolution (CR); and
3. **Preprocessing on KB.** For the textual information in the KB (e.g. description), we apply: TOK, NER, and CR;

Linking Approach

First, we normalize the input mention string, followed by querying the KB with the resulted mention string that will output a list of KB unique identifiers. In order to pick the best KB unique identifier, we train an SVM model on the gold annotated dataset using a set of features that includes:

Rank Feature For every query mention string, we use a popularity-based ranking algorithm that is based on the count of inbound and outbound links in KB.

String Similarity Features Such as: a) the longest common substring between the surface forms on the query and the mention in the KB; and b) The overlap with the aliases as defined in Equation 3.

$$MaxOverlapAlias_j = Max_j(S_i) :$$

$$S_i = Max(substring(alias_{ji}, queryStr_j)), \\ i \text{ in } \{1, \dots, \#aliases\}, j \text{ in } \{1, \dots, \#queries\} \quad (3)$$

Entity Type Features We mainly focus on the three conventional entity types: Person (PER), Location (LOC), and Organization (ORG). We generate the entity type for both the textual data *Type-doc* and the KB data *Type-kb*. For the textual data we use the output of the Stanford NER system (Manning et al., 2014a). On the other hand, we built an in-house multiclass SVM classifier (Vapnik and Vapnik, 1998) for *Type-kb* that uses string features in addition to: set of inbound and outbound relations that are connected to the mention and gazetteers (PER, LOC, ORG).

Table 12, depicts the Precision (P), Recall (R), and F1-measure of our classifier on each of the three classes.

Table 12: Entity Type Classifier Results

	P	R	F1-score
PER	92.5	100	96.1
LOC	86.4	69.5	77
ORG	96.3	90.1	93.1

Gender Features Gender is helpful in disambiguating person entities (example is shown in Figure 7). Gender feature values are {Female, Male, Other, NA}, where the values are generated based on the coreference cluster ID. For instance, if we have the following sentence: *Obama* is going to make his speech after *he* meets with Kerry. *Obama* and *he* both belong to the same coreference cluster, thus Obama gender is assigned the *Male* label. Gender is generated for both textual and KB sides, where we use graph relations for the latter.

Semantic Features : We add semantic features by modeling the overlap of lemmas between the textual data in the documents and the KB. Additionally, we think that the named entities that occur in both texts can provide important clues in linking and disambiguation, and we therefore add information regarding the overlap between NEs as a feature.

4.2.1 Datasets & Tools

Datasets

For this task we use the datasets provided by TAC-KBP 2015 for EDL task. The dataset comprises:

- Freebase dump (LDC2015E42);
- Training data (LDC2015E75) which contains the gold mentions with the document, offset, entity type, and link ID or NIL cluster ID in case there is no entry for the input mention, in addition to the source documents where the input mention occurs in. The training data contains Chinese and Spanish data in addition to English, however we only report results on English name mentions; and
- Test data (LDC2015E102) which contains the source documents of the input queries along with the query file without the gold annotations. Table 13 reports the training and test data stats.

Table 13: Entity Linking and Clustering train and test data stats

	Train			Test		
	NW	DF	Total	NW	DF	Total
EN & NAM	4811	7369	12180	6377	7664	14041
num_documents	83	85	168	83	85	168
num_NILs	712	2557	3269	2260	2522	4782

The data contains both newswire (NW) and discussion forums (DF) genres, where the type/token ratio (TTR) in the training and test data is about 20% in NW, and about 15% in DF.

We build one model for DF and NW data using SVM, despite the fact that we have more DF data, hence the training models will be more skewed.

Tools

The following tools are used:

1. standalone Apache Jena Fuseki server¹ (SPARQL server);
2. Stanford CoreNLP (Manning et al., 2014a). We use this toolkit for the following tasks: Tokenization, POS tagging, Coreference Resolution, Named Entity Recognition (NER); and
3. WEKA (Hall et al., 2009). We use WEKA for classification and clustering.

4.2.2 NIL Clustering approach

For every input mention, we create a feature vector that encompasses the following: a) Entity Type; b) Type of document (newswire vs. weblogs); c) Coreference cluster ID concatenated with the name of the file, which would be helpful in the case of resolving intra-document entities; d) Part of Speech (POS) tag; e) sentence2vec: This feature uses the word vector representation of the words W in the sentence S and calculates their average, as shown in Equation 4. Then, we use K-Means clustering (Hartigan and Wong, 1979) for grouping NIL (no match with KB entries) instances. We experiment with multiple number of clusters $k = 1000, 1500, 2500, 3500, 4000$, where k is empirically chosen.

4.3 Results and Discussion

Results

Table 14 depicts our system results for Precision, Recall, and F1-score for English. To evaluate the linking results, we adopt the following measurements² as defined by TAC-KBP:

- `strong_link_match`: micro-averaged evaluation of links. A system link must have the same KB gold link identifier to be counted correctly;
- `strong_linked_mention_match`: measures non-NIL mentions that are linked to KB identifier, regardless of correctly linking to the right entry in KB;
- `strong_nil_match`: micro-averaged evaluation of NIL mentions; and
- `strong_all_match`: micro-averaged link evaluation of all mentions. A mention is counted as correct if it either represents a link or NIL match.

Table 14: Precision (P), Recall (R), F-score (F1) of GWU team at TAC-KBP

measure	P	R	F1
strong_link_match	77.5	37.8	50.8
strong_linked_mention_match	95.9	46.7	62.8
strong_nil_match	36.5	73.8	48.8
strong_all_match	52	48	49.9

Based on the results in Table 14, we notice that in evaluating the linking performance (strong_link_match), the precision (77.5%) is much higher than recall (37.8%). This is a result of our proposed system that focuses on maximizing precision as opposed to recall; where the latter requires query expansion and augmentation techniques. This is based on our goal of using the linking information in our aliasing system solely in high linking confidence instances.

$$sentence2vec(S) = (\text{SUM}_i^{\#words} W_{1i}, \dots, \text{SUM}_i^{\#words} W_{di}) : d = [1, dimension] \quad (4)$$

4.4 Conclusions

In this chapter, we explained and discussed our English entity linking and NIL clustering system that represents a preamble to our NEAR system described in Section 5.

¹https://jena.apache.org/documentation/serving_data/

²<https://github.com/wikilinks/neleval/wiki/Evaluation>

Chapter 5

Named Entity Alias Resolution (NEAR)

5.1 Introduction

Named Entity Recognition is the initial step for a robust information extraction system, that later involved entity linking and clustering tasks. Entity disambiguation and clustering, which we will use the term Named Entity Aliasing resolution interchangeably, is more focused on the contextual and semantic aspect of the problem, where the goal is to group/disambiguate different mentions of the same named entity in one/multiple group(s).

In this chapter we will cover the aliasing resolution task and approaches. We start by covering our approach in the area of Named Entity Aliasing Resolution (NEAR), with a focus on personal named entities (PNEAR), and discuss our results on two different languages, namely Arabic and English. Additionally, we propose our pseudo-generated Arabic aliasing dataset.

It is worth noting that the motivation behind focusing on unknown entities is a result of our interest in developing techniques that can target low-resource languages, especially that they are more likely to suffer from knowledge bases with low coverage and recall. Furthermore, the need to target social media is becoming more crucial where new entities are coming to life at a rapid rate, and the need for unsupervised and language-independent approaches is essential given the high cost associated with annotation in general, and the extra layer of difficulty that this task imposes, in specific, due to its subjectivity.

Our work in this domain resulted in the following contributions:

- Introduce aliases dataset for MSA that is targeted for aliasing resolution task; and
- Aliasing resolution model that produces state-of-the-art results for both aliasing resolution for English entity disambiguation task and Arabic.

5.2 Methods, Assumptions, and Procedures

5.2.1 Current Aliasing Resolution Approach

We use the term Aliasing Resolution interchangeably with NIL clustering and cross-document coreference. In our work, we focus on the true-aliasing instances that are not easily identified using string edit distance measurements, which represents a more challenging task.

Our current framework is composed of two main components: i) Extracting features from text and generating a feature vector for every NE, where the resulting vector space model is a Name Features Relatedness (NFR) matrix; ii) Identifying the groups of aliases using two clustering algorithms, namely, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996) and optimized K-means clustering (Hartigan and Wong, 1979). In this approach we explore PNEAR in a multilingual setting for English and Arabic languages and report results on both. We use both monolingual and cross lingual features in these systems.

Extracting NE features

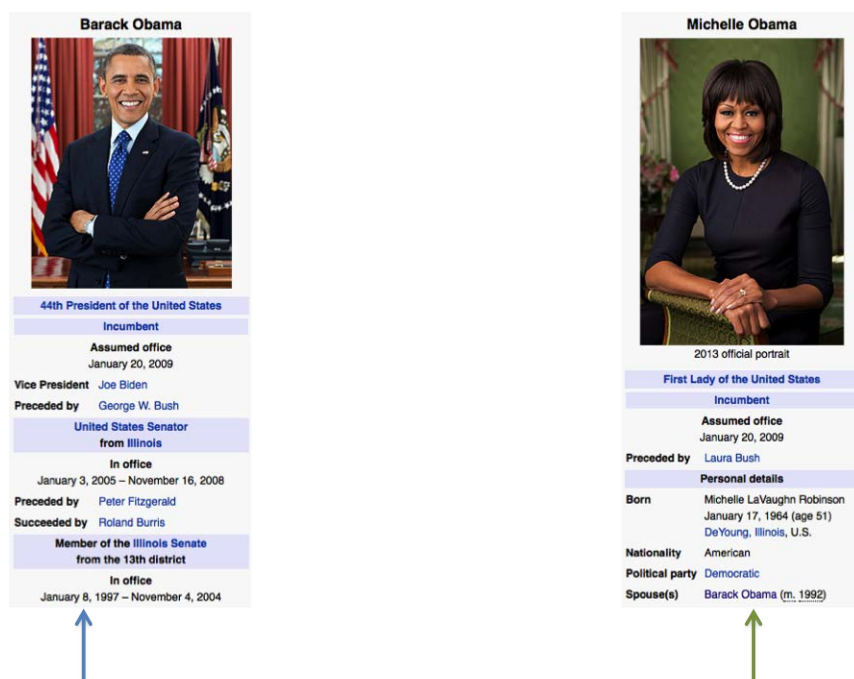
Orthographic String Similarity Features

Monolingual Features To identify misspelled names, name mention variations or transliterated aliases, we use the normalized Longest Common Subsequence (LCS) metric, as shown in Equation 5 to measure the string similarity between the two names (Hirschberg, 1977).

$$\text{NORMLCS}(\text{name1}, \text{name2}) = \text{LCS} / \text{max length}(\text{name1}, \text{name2}) \quad (5)$$

Cross-Lingual Features We observe that two transliterated names will have a higher probability of mapping to the same name in another language (e.g. "byl klyntwn" and "byl klntwn"¹ will both map to "Bill Clinton" in English). As a result we opt to use the translation of the names as an indicative to transliteration (Zirikly and Diab, 2013). Other features used in our system are:

Gender: We use the gender feature as proposed in Chapter 4.



Barack Obama visited Germany with **Michelle Obama** to meet Merkel. **Obama** made agreement with the German government regarding some economical issues.

Figure 7: Entity Clustering Example

Shallow Syntactic and Semantic Features:

- **Monolingual Part of Speech Features (POS)** We use shallow syntactic information (POS) tags within a window as features based on the observation that similar entities are generally mentioned in similarly structured sentences. To reduce the sparsity of POS tags, we map POS tags to five different coarse labels: Noun, Verb, Adjective, Adverb, and closed-class for all other tags; and,

¹Arabic is encoded using the Buckwalter transliteration scheme as described in <http://www.qamus.org/transliteration.htm>

- **Semantic Role Labeling Features (SRL)** SRL provides the semantic argument and role associated with NE in a sentence. We directly incorporate SRL features generated for English using a semantic parser. However, due to the lack of SRL systems for Arabic, we rely on projecting the semantic role labels of the NE from parallel English resources tagged using an English SRL system on corresponding Arabic data where the corresponding English NE translation token is identified.

Latent Named Entity Topics (NE-TOPIC) We experiment with topic models to discover the latent relations between the named entities. We apply Latent Dirichlet Allocation (LDA), as described in (Blei et al., 2003b) to reveal the underlying topic dimensions pertaining to each of our named entities. We create the models on both English and Arabic data respectively.

Co-occurrence Context Features (COF) The context vector will depict the frequencies of all the words that co-occur with the alias within a predefined context window (= size of NE’s sentence).

Similarity Metrics and Clustering

Feature Vectors Similarity The similarity function between two feature vectors used in our clustering algorithm is shown in Equation 6, where: NE1, NE2 are the two named entities of interest, $\alpha = prob(NE1, NE2)$ shown in Equation 7 represents the probability that NE1 and NE2 co-occur in the same document, $\cos$ is the cosine similarity function, $LCS(NE1, NE2)$ is the LCS value of the two names, $LCS-TRANS(NE1, NE2)$ is the LCS value of the translated names, GEN is a gender binary feature that is set to true if the two entities’ gender are the same.

$$\begin{aligned} sim(NE1, NE2) = & \alpha \times SUM \cos(NE1.cof, NE2.cof) \\ & + \cos(NE1.pos, NE2.pos) + \cos(NE1.srl, NE2.srl) \\ & + \cos(NE1.NE-TOPIC, NE2.NE-TOPIC) \\ & + LCS(NE1.name, NE2.name) \\ & + LCS-TRANS(NE1, NE2) + GEN(NE1, NE2) \end{aligned} \quad (6)$$

$$prob(NE1, NE2) = (|D_{NE1} \cap D_{NE2}| / |D_{NE1} \cup D_{NE2}|), D_{nei} : \text{Documents contain } nei \quad (7)$$

NEs Clustering

We empirically investigate different clustering algorithms, where K-means emerged as the best performer. To avoid using a predefined k , we use DBSCAN to learn the best number of clusters based on the density of the NEs in the documents of interest. We apply DBSCAN on the NE-TOPIC matrix and empirically tune (the neighborhood maximum distance threshold) with a range of values based on the mean of the pairwise distance among our data points. Secondly, we employ the locally optimized K-means clustering algorithms² that use k produced by DBSCAN and the similarity function discussed previously to generate the groups of aliases.

5.2.2 Datasets & Tools

Datasets

Most of the datasets that are relevant to the entity clustering and aliasing resolution task either focus on the named entity disambiguation or the coreference resolution tasks in English. Notwithstanding, there have been multilingual datasets (McNamee et al., 2011) that handle Arabic, they suffer from the problem that annotation was not capable of capturing the true-aliases cases (an example from this dataset is Abu Mazen

²We use the Repeated Bisection clustering method, a flavor of k-means.

as an alias of Mahmoud Abbas). Thus, we conduct two sets of experiments on two different datasets: (i) “John Smith” dataset (Bagga and Baldwin, 1998), which includes 197 mentions of John Smith adopted from The New York Times articles mapping to 35 unique entities, that would allow us to handle the entity disambiguation aspect of the problem; and (ii) Our own created dataset that is adopted from the Gale parallel dataset for English & Arabic that contains true-alias instances.

On the Arabic data set we perform the following annotation preprocessing steps:

- Generate a normalized list (e.g. Alef normalization) of Personal Named Entities (PNE) after running Named Entity Recognition (NER) system on the dataset (the data is the GALE Phase II Release1 parallel data for English & Arabic.³);
- Manually identify transliterated, misspelling and mention variations cases and group them in clusters (resulting in 12014 documents);
- For every person named entity, generate two pseudo random names that are added as aliases to the entity using Fake Name Generator ⁴, resulting in the gold aliases list (L); and
- Replace every occurrence of a name (n) in the previous list (L) with a randomly selected member of the entity where ($n \in L$), preserving the original frequency distribution of the entity. The gold list L comprises 96 names grouped in 30 unique PNE clusters (Zirikly and Diab, 2013).

For evaluating the quality of clustering, we use the B-CUBED scoring algorithm proposed in (Bagga and Baldwin, 1998). This metric is used as the official metric in the TAC-KBP entity clustering task evaluation.

Tools

- Arabic Preprocessing: we use MADAMIRA (Pasha et al., 2014) to generate POS tags, lemmas, gender, and the English translation of names;
- Arabic NER: we use an in-house built NER system that is integrated with MADAMIRA;
- English Preprocessing and NER: we use Stanford CoreNLP (Manning et al., 2014a) to generate the same lists of features as 5.2.2 and 5.2.2; and
- Clustering: MATLAB implementation of DBSCAN⁵, and CLUTO implementation of the Repeated Bisection clustering algorithm (Karypis, 2002);

5.3 Results and Discussion

Baselines

The state-of-the-art results for the “John Smith” dataset are 92.02% F1 (Chen and Martin, 2007), which we compare our results to. On the other hand, we proposed two baselines for our Arabic dataset: the first uses two variations of Longest Common Subsequence (LCS): i) LCS on the names in the Arabic dataset yielding an F1-score=37.67; and ii) LCS-TRANS on the translated names in English yielding an F-score=40.37. The second baseline adopted is a random assignment of each of the 96 different names to the $[0, k]$ clusters (where k is learned from DBSCAN), which achieves an F-score=34.98. Some might argue that the baselines proposed for our dataset are weak, however string similarity measurements have been commonly used as baselines in transliterations extraction tasks (Bergsma and Kondrak, 2007). Additionally, we propose the α weighted BOW model as another baseline that captures the true variant cases.

³LDC2007E103. <https://www ldc.upenn.edu>

⁴<http://www.fakenamegenerator.com/order.php>

⁵<http://www.mathworks.com/products/matlab/>

Results

In our experimental study, we use the lemmas (English) and diacritized lemmas (Arabic) when generating the BOW and PNE-TOPIC matrices. Table 15 illustrates our system results using orthographic, syntactic, and semantic features, where our proposed compound model (including all the features: LCS, LCS-TRANS, a reduced version of BOW that uses Brown clustering (Bag of Browns, BOB), PNE-TOPIC, SRL, POS) shows 24.6% improvement over the use of orthographic features alone.

Additionally, lexical semantic features, latent topics namely PNE-TOPIC, when employed alone, also yield improvement in performance over simple orthographic features. Using Brown cluster IDs, instead of the actual representation of words, as features results in an improvement of about 19% over the LCS baseline. We pick a window-size = 1 for POS as it seems to slightly outperform a window size of 2.

Using SRL or POS features alone yields, as expected, the lowest performance, since it conveys a lower information value due to the high frequency of these features. Overall the best performance is produced by the compound model. Although α -weighted values of the compound system boost the system performance minimally, the results strongly suggest that our hypothesis about the higher alias likelihood of names that co-occur in the same document is sound.

In Table 15, due to the different characteristics of “John Smith” dataset, we notice myriad differences among the different features. First, we note that the compound framework fails to deliver the best performance in comparison to PNE-TOPIC. This is due to the poor performance that BOB yields, as a result of the structure of the dataset where every mention of *John Smith* occurs only once, thus, the singleton cases will have a small feature vector (i.e. a large number of zero-value features) as opposed to the denser feature vector for *John Smith* instances that occur more often (An example from the data of the size percentage of a singleton feature vector vs. dense feature vector: 24.83%). The α value in the “John Smith” data is equal for all the John Smith instances, because every document contains only one occurrence of *John Smith*.

Table 15: F-score PNEAR results with the use of different features

Authors created dataset	
Feature	F-score
RANDOM	34.98%
LCS	37.67%
LCS-TRANS	40.37%
Alpha weighted BOW model	61.05%
POS-pos -window = 1	42.82%
POS-pos -window = 2	37.95%
PNE-TOPIC, num-topics=200	45.96%
SRL	31.6%
BOB	60.23%
Compound model, pos -window = 1	64.84%
Compound model, pos -window = 2	64.35%
Alpha weighted compound model, pos -window = 1	64.97%
“John Smith” dataset	
Feature	F-score
BASELINE	92.02%
BOB	72.56%
PNE-TOPIC, num-topics=200	96.57%
Compound model	75.09%

We conducted a qualitative error analysis on the performance of the proposed framework on our created dataset, as it contains transliterations and true-alias cases. We note that our framework performance on the transliterated names where LCS and LCS-TRANS are the only used features, yields, as expected, a higher performance of 72.58%. However, we could not train the system on a threshold of orthographic weights vs. other features, especially since the dataset is too small which might cause over fitting, thus we use our uniformly weighted compound system. We note that our compound system produces a higher performance when tested solely on the true variant cases with an F-score=69.69%. This is a result of the syntactic and semantic features dominance over the orthographic features. It is also worth noting that using an α weighted BOW model yields an overall performance of 61.05% on the aliasing problem, where an improvement of approximately 4% has been achieved when using our weighted compound framework. Hence our approach of using denser features yields superior performance on the harder problem of complete name variants. In the “John Smith” dataset, the singleton cases are mostly the reason behind affecting the performance where these cases are mainly assigned to the cluster ID representing the most occurring instance of John Smith.

Table 16: PNEAR System performance when tested on Transliteration vs. Variant cases

Alias type	% Eval Data	LCS&LCS-TRANS	<i>compound system</i>
Transliterations	43.75%	72.58%	55.16%
Variants	56.25%	51.36%	69.69%

5.4 Conclusions

In this chapter, we introduced our probabilistic framework to perform the aliasing resolution task with a focus on disambiguation and true alias cases. The model developed generates state-of-the-art results for both English and Arabic, and emphasizes the need to have datasets that focus on true alias instances that cannot be mapped to entries in a KB, which will result in more robust information extraction systems.

Part II

Cognitive State

Chapter 6

Belief Word Tagging

6.1 Summary

In this section we illustrate the use of Deep Neural networks for modeling committed belief tagging (identifying the *level committed belief* or LCB). We have two separate models, one for committed belief toward propositions and one for tagging committed belief toward events. The proposed model yields an F-measure of 76.15% which is an improvement on previously published models by 69.1%. In another task and a shared pipeline, we explored the problem of LCB toward events (LCB-E) but on a smaller dataset with fewer LCB annotations and on a specific set of events. We examined the impact of multitask learning frameworks (MTL), along with the writing style classification (WSC) task on the LCB-E task.

6.2 Introduction

Identifying the level of committed belief (LCB) of a speaker/writer (SW) is of relevance for opinion mining, veridicality and certainty assessment of one’s cognitive state (Wiebe et al., 2005). This modality of natural language has been used to forecast political opinions of groups, their level of commitment in a proposition could be indicative of their political decisions. LCB also contributes to identifying people’s attitudes towards entities or events; are they hedging or committed to a certain view about a topic/event/entity? Also, LCB may help in calibrating the level of confidence we have on how people express their opinions.

In this paper, we present a novel deep learning model for learning and tagging LCB-P and LCB-E using two different data sets of Linguistic Data Consortium (LDC). We first follow work by (Werner et al., 2015b). But different from (Werner et al., 2015b), we do not rely on hand crafting features, we adopt a feature representation approach. We propose a deep learning model using a recurrent neural network architecture for the LCB-P task which is experimented on proposition level and a relatively larger dataset of discussion forums (DF).

We then conducted the LCB-E task on event level using a smaller dataset of LDC, collected from discussion forums and newswire (NW) data. It is driven by our intuition that writing style is indicative of level of commitment. Hence, we studied LCB on DF and NW and implemented a hard parameter sharing MTL (HMTL) to efficiently and quickly perform LCB-E and WSC tasks in a single unified model. Our approach is supervised for LCB-E as we leverage manual annotations in the standard dataset comprised of English DF and NW data provided by LDC. For WSC data, we designed a distance supervision model (DSM) that we learned the writing style from the data format and place holders. We adopted a weakly supervised framework, where we assumed the data genre to be indicative of the formality level and writing style of the author of the text. We assumed that if a text comes from a news headline, then the style would be different from DF. And finally, we devised a classification task where the objective is to identify the genre class of the text: NW or DF.

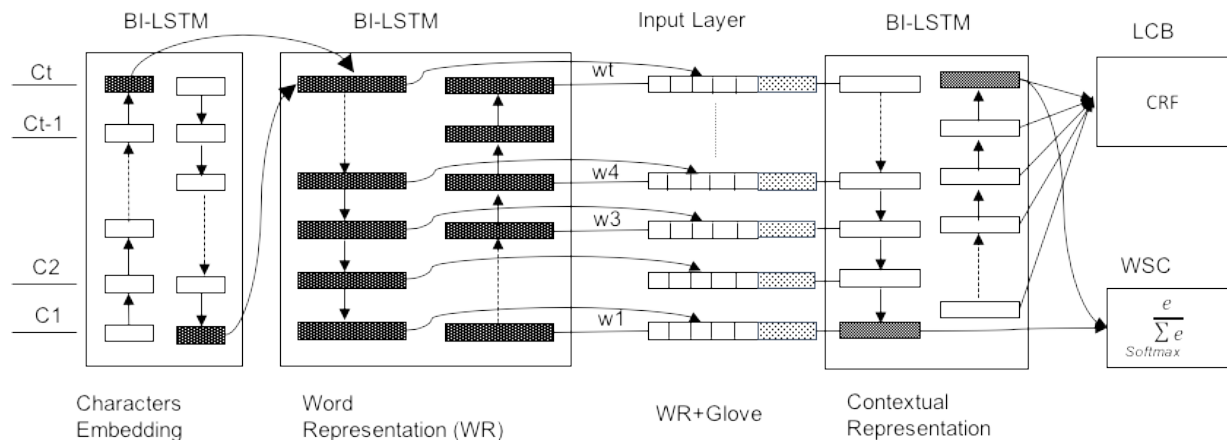


Figure 8: Multi-task hierarchical bidirectional LSTM learning model for LCB and WS/genre classification

6.2.1 Background

The initial work on identifying the level of committed belief (LCB) and factuality was published in (Diab et al., 2009) (M9) releasing the LU corpus. The LU corpus comprises the annotation of propositional belief in context. It was devised as three levels of annotations of LCB: committed belief (CB) where the writer strongly believes in a proposition; non-committed belief (NCB) where the writer reflects a weak belief in the proposition; and non-attributable belief (NA) where the writer is not (or could not be) expressing a belief in the proposition (e.g., desires, questions, etc.). Later on, (Saurí and Pustejovsky, 2012) contributors, proposed a more fine-grained annotation scheme merging the Factbank and the LU corpus (Prabhakaran et al., 2010b) as well as (Prabhakaran et al., 2015b) (P15). They added another tag reported belief tag (ROB) which annotates the case where the writer’s intention is to report on someone else’s stated belief, regardless of their own belief. For our purposes, we adopt the latter 4-way annotation levels for the LCB task, namely we have a 5-way (CB, NCB, ROB, NA, and NO) classification task at the token level.

6.3 Methods, Assumptions, and Procedures

6.3.1 Approach

In this section, we first introduce the base neural architecture for the LCB-P task as it is shown in Figure 8 and then we discuss the HMTL framework used for LCB-E and WSC tasks. Considering the MTL definition (Caruna, 1993) (Yang and Hospedales, 2016), we aim to improve the learning of the model by leveraging the homogeneous tasks with a hard parameter sharing pipeline.

6.3.2 Level of Committed Belief on Proposition (LCB-P)

The goal of the LCB-P task is to identify the level of committed belief of SW toward the head of proposition.

Exploiting the same LCB categories as (Prabhakaran et al., 2010b) and M9 we came up with **Committed belief (CB)**, **Non-committed belief (NCB)**, **Reported belief (ROB)**, and **Non-belief (NA)** toward the head of proposition, Examples:

(CB) Congress will **vote** for the new tax law next week.

(NCB) He could **win** the coming presidential election.

(ROB) CNN reported he could **win** the election.

(NA) Is she going to **win** the election? I hope people do not **vote** for him.

Approved for Public Release; Distribution Unlimited

(Prabhakaran et al., 2015b) and G10 relied on handcrafted features, despite the importance of such a technique it is not extensive when it comes to complex multigenre domains. In this problem, we exploit the word representation which is sensitive to the spelling and novel words. We construct the character and word representation alongside the pre-trained word embedding (Pennington et al., 2014), which captures the morphology, overcomes the novel words within context, and has a sound cross domain capability. The impact of using the dense representation and LSTM networks to capture long dependencies within context has been widely discussed in sequence tagging problems like part of speech (POS) and name entity recognition (NER). So called LSTM-CRF leads to state of the art results in NER (Lample et al., 2016a). We employed the LSTM-CRF pipeline as shown in Figure 8. The proposed graph for LCB-P and LCB-E includes two layers of bi-lstm to construct the word embeddings from characters. The word embedding representations are concatenated with a pre-trained lookup embedding output of each word. The vector representation of each word is fed as an input layer to the graph with a bi-directional LSTM layer (Bi-LSTM). For better generalization and improving the performance, we added a dropout layer after the input layer to make sure our model was not dependent on any of the representations. Instead of using the softmax layer we used the CRF classification layer to take into account the adjacent tokens in sequence.

LCB tagging Scheme The task of LCB-P and LCB-E is assigning a belief label to every word in a sentence. A single belief tag could be spanned to more than one token, therefore we use the IOB scheme where every token is labeled as B-CBlabel if the token is the beginning of CB, I-CBlabel if it is inside CB, or O-CB otherwise.

6.3.3 Multi Task learning for LCB-E and WSC

In this experiment, we employed the hard parameter sharing MTL (HMTL) to perform LCB-E and WSC tasks in a unified pipeline.

Task 1-LCB-E

Our goal is to detect the level of expressed belief held about certain events instead of the head of a proposition which was discussed in LCB-P. We consider the event as something that happens at a particular place and time and corresponds to only eight types of events: life, movement, business, conflict, contact, personnel, transaction, and justice events. The following example indicates the LCB-E and LCB-P in a single sentence;

He asked **(CB-E)(CB-P)** my sister to go **(NA-E)(NA-P)** to the party **(CB-E)**.

The go event is labeled NA because, even though the SW is committed to the belief that his sister was asked to go to the party, he is not expressing a belief about whether or not she is going to the party. The party event is labeled CB because the SW made certain that the party exists (party is not labeled as CB-P since it's not the proposition). As it is shown, the LCB-E task is a more challenging task since it demands the identification of events and deep interpretation of the context.

Task 2-Writing Style Classification (WSC): Language consists of much more than just content (Pavlick and Tetreault, 2016), choice of word, entities, hedging and many other elements play a key role in changing a tune of a sentence. Looking at following two sentences:

(NW) -*Jeffrey S. Robbins the chair of the board of the New England Anti Defamation League, was among those who testified (cb-B) in support of Berman at his nomination (cb-B) hearing last week.*

(DF) -*He said (cb-B) he would meet (na-B) dangerous world leaders for top level talks without preconditions if elected (na-B) president.*

Both sentences are selected from the LDC belief and sentiment dataset, but the first is more formal. The formality of a sentence has a tight relation with the level of belief. Considering the data domain, we came

Table 17: Training and evaluation set statistics

Annotation	CB	NCB	ROB	NA
<i>DF 2014 Training Set</i>				
143,240	79995 (56%)	3890 (3%)	7150 (5%)	52205 (36%)
<i>DF 2014 Evaluation Set</i>				
17,553	8730 (50%)	583 (3%)	941 (5%)	7299 (42%)
<i>DFNW 2015-17 ERE Training Set</i>				
10455	8189 (79%)	171 (1%)	73 (0.6%)	2022 (19%)
<i>DFNW 2015-17 ERE Evaluation Set</i>				
1563	1208 (77%)	16 (1%)	7 (0.4%)	332 (21%)

Table 18: Training and evaluation set statistics for WSC task

DFNW Training Set					
	cb	ncb	rob	na	WSC
DF	2703	80	32	818	12139
NW	1157	12	11	99	1188
DFNW Evaluation Set					
DF	635	10	3	179	2563
NW	230	2	2	36	347

up with two classes with the following tags: DF and NW. WSC is a binary classification task to detect the writing style of a given document. We propagate these labels to the sentence level thereby acquiring labels in a weakly supervised manner using the distance learning technique. We used the same shared pipeline as LCB-P which is shown by the dotted area in Figure 8. For each sentence, we use the output states containing the forward and the backward final states of a bidirectional RNN as the input to the softmax layer.

6.3.4 Experiments and Evaluation

In this section, we explicitly discuss about the dataset, experimental setup and results of each component.

Dataset

We employed the LDC DEFT Belief and Sentiment dataset (collected and manually annotated between 2014 and 2017). The source of data, annotation pipeline, and further description are explicitly explained in (Prabhakaran et al., 2015b) and (Rambow et al., 2016). To our knowledge, the LDC 2014 (DF14) dataset is the largest dataset available for belief toward proposition (DF14 is collected from English discussion forum threads). LDC 2015-17 ERE (DFNW) is the Belief-Sentiment dataset which all source documents are collected from English discussion forum threads and newswire data and the belief annotation marks the belief-holder’s commitment to a belief in the occurrence of specific events (event-target) and entities (entity-target) but we only used the event-target annotations as the ground truth for the LCB-E task. We split each of the DF14 and DFNW datasets into 90% train and 10% development, and after achieving the best performance on the development set, used the given evaluation set for the final result. Table 17 shows the overall stat of the belief tags and some statistics about the train and evaluation corpus including the distribution of each tag in the dataset. Table 18 illustrates the stat of DF and NW in NWDF and also the distribution of LCB-E tags in each WSC categories.

Table 19: Results obtained for the LCB-P and P15 (of (Prabhakaran et al., 2015b)) model on the LDC 2014 Evaluation dataset.

	P15			LCB-P		
	Pr	Re	F1	Pr	Re	F1
CB	68.9	77.9	73.1	75.15	84.55	79.57
NCB	52.9	29.7	38.0	57.02	50.78	53.72
ROB	43.8	15.6	23.0	53.98	33.89	41.64
NA	80.1	62.0	69.0	82.00	73.28	77.39
Overall	72.0	66.4	69.1	76.35	75.95	76.15

Experimental Setup and Training

For all the pipelines, we trained the network with back propagation and used the Adam (Kingma and Ba, 2014), RmsProp and SGD optimization algorithms. We updated our parameters in every training epochs. The character and Glove pre-trained embedding size [100,200,300] are examined with batch size 20 with 100 epochs. The highest performance on the development set was achieved under the following parameters: hidden size of Bi-LSTM (100); optimization (RMSprop); initial learning rate (0.003); L2 ($Landa = 0.1$), character and word embedding size (300,100); dropout (0.3).

6.4 Results and Discussion

We implemented two main experimental pipelines: LCB-P and multi task learning pipeline for LCB-E and WSC tasks. We conducted our experiments in two different phases. In the development phase, we utilized development data for tuning. In the evaluation phase, the model with the highest performance in the development phase was used to test the evaluation set. Evaluating the performance of the proposed techniques on LCB-P, LCB-E and WSC relied on the number of relevant tags that are selected (recall) and the number of selected belief and writing style tags that are relevant (precision), since both of them are presented in this work.

LCB-P Table 19 presents the comparison of the proposed LCB-P model with p15 results. In order to have a fair comparison we used the same experimental setup as (Prabhakaran et al., 2015b). Their model employed a lot of hand-crafted features such as hedging, part of speech, dependency parser, sentiment, and WordNet. The proposed LSTM-CRF model could outperform the previous models in LCB-P tasks. Table 19 presents the overall results tested on the evaluation set in comparison with p15 results. We could improve the overall recall and precision to 75.95 and 76.35 respectively, and reaching to 76.15 f-measure in comparison to 69.1 reported by (Prabhakaran et al., 2015b).

LCB-E and WSC Table 20 presents the overall performance of different experiments on the LCB-E and WSC tasks. We have chosen multiple baselines for this experiment. First the MIX experiment exploits the mix cross domain training sets to detect the LCB-E. In this experiment, we combine the DF14 and DFNW training sets for training and used the HMTL pipeline for training. In another experiment, we used the LCB-P prediction model to perform the evaluation. The LCB-E task presents the overall performance of the LSTM-CRF pipeline on the LCB-E tasks. The MTL tasks demonstrate the performance and the impact of adding WSC and LCB-E in the HMTL pipeline. We also assess the LSTM-Softmax pipeline with single task for the WSC task.

6.5 Conclusions

In this chapter, we proposed and evaluated a general-purpose pipeline for Committed Belief detection toward proposition and event. We further assessed the influence of writing style on the LCB task and improved the overall performance of WSC and LCB-E tasks in HMTL.

Table 20: Results of employing MTL network on LCB-E and WSC tasks

Task		WSC			LCB-E		
		Pr	Re	F1	Pr	Re	F1
LCB-P		-	-	-	14.17	27.87	18.78
LCB-E		-	-	-	52.18	30.79	38.72
MIX		-	-	-	45.30	26.41	33.36
MTL	DF	93.68	96.66	95.14	46.45	40.69	43.37
	NW	70.67	61.99	66.04			
WSC	DF	94.40	95.36	94.87	-	-	-
	NW	60.48	58.21	59.32			

Chapter 7

Sentiment

7.1 Summary

We have developed methods for identifying sentiment in social media and news. Our focus early in the program was on the development of methods for identifying sentiment in social media, given that most previous work had been done for news. We experimented with features that are useful for social media and used domain adaptation methods. Our work at this phase was in English. Using these approaches, augmented with methods to find source and target (see Chapter 8), we participated in the 2013 and 2014 TAC sentiment slot filling evaluations. We also participated in the SemEval evaluations on sentiment and the primary PhD student for this work was asked to join the organizing committee for the SemEval track on sentiment. In later phases of the project, we developed a sentiment system for Spanish twitter and participated in TASS 2015 Workshop on Sentiment Analysis at SEPLN. In the final stages of the project, we developed a new sentiment analysis system for English using word embeddings that we showed was state of the art and we developed new more sophisticated methods for identifying source and target. We ported the English approach to both Spanish and Chinese and we participated as part of the Tinkerbell team in ColdStart++.

7.2 Sentiment Analysis for Social Media

7.2.1 Introduction

We developed a novel supervised sentiment detection system that classifies the polarity of subjective phrases as positive, negative, or neutral (Rosenthal and McKeown, 2013). It is tailored towards online genres, specifically Twitter, through the inclusion of dictionaries developed to capture vocabulary used in online conversations (e.g., slang and emoticons) as well as stylistic features common to social media. We show how to incorporate these new features within a state of the art system and evaluate it on subtask A in SemEval-2013 Task 2: Sentiment Analysis in Twitter.

7.2.2 Methods, Assumptions, and Procedures

We extended this approach to participate in the slot filling sentiment task run by TAC (Sara Rosenthal, 2014). Through our participation in the Sentiment Slot filling Task of TAC KBP 2014, we developed a system to address the challenges of identifying pairs of entities related by sentimental expressions, both in newswire text and online discussion forums. It builds on our 2013 approach which identifies named entities and entity mentions, establishes grammatical relationships, and determines the polarity of any sentiment that may exist between them. In the 2014 TAC task, we have additionally built a sentiment knowledge base which is used as a prior polarity score for the candidate entities for the query.

We also entered these sentiment systems in Subtasks A and B of SemEval2014 Task 9: Sentiment Analysis in Twitter (Rosenthal et al., 2014).

Table 21: Results for the TAC 2014 task

Run	First Mention	Threshold	Knowledge Base	Precision	Recall	F-Score
1	Yes	None	Yes	7.1%	17.0%	9.9%
2	Yes	None	No	7.1%	17.0%	9.9%
3	Yes	.75	Yes	9.8%	8.3%	9.0%
4	Yes	.75	No	9.6%	9.2%	9.5%
5	No	None	Yes	6.8%	20.7%	10.3%
5	No	.50	Yes	18.3%	6.6%	9.8%
5	No	.75	Yes	12.9%	8.5%	10.2%

Table 22: Results in the SemEval2014 evaluation (Subtask B)

Experiment	Twitter			SMS	LiveJournal
	Dev	Test	Sarcasm		
Majority	29.2	34.6	27.7	19.0	27.2
Phrase-based System	62.1	64.7	40.8	56.9	62.2
Tweet-Level System	62.4	63.7	42.4	60.5	69.4
Combined System	64.6	65.4	40.0	59.8	68.8

7.2.3 Results and Discussion

Our results for the 2014 TAC task are shown below in Figure 21. While these are low F-scores, it should be noted that most systems perform poorly on this task because it requires correct identification of three different factors in the result (source, target and sentiment). In our analysis of results, we found that the greatest source of error came from the detection of entities for source or target. We used Serif for this task and in some cases, it did a poor job of entity detection. However, in other cases the problems were caused by the assignment of an entity with type location to the source, for example. Incorrect assignment of an inanimate entity to a role that must be animate also caused problems. In general, detection of sentiment itself was much more accurate.

Our results in the SemEval2014 Subtask B are shown in Table 22. We experimented with different methods and the combination of all methods did the best. Overall in SemEval, we came in 15th out of 50 participants.

7.2.4 Conclusions

Our work on sentiment analysis resulted in approaches for detecting sentiment in a variety of online genres. We extended that to address identification of sentiment relations between source and target. We evaluated our work in TAC KBP and SemEval. Our results in SemEval show that we are doing comparatively better than most other approaches.

7.3 Sentiment Analysis for Spanish Twitter

7.3.1 Introduction

The vast majority of work on sentiment analysis has been on English texts. Since methods for determining sentiment often rely on language-specific resources such as sentiment-tagged thesauri, they are often difficult to adapt to languages beyond English, as other language often have scarcer computational resources.

Table 23: Sentiment Analysis Results at a Global Level

Task	Test Set	Variant	Acc.	Prec.	Rec.	F1	System Rank	Group Rank
5 Labels	Full	Six-way Composite	0.50 0.36	0.39 0.31	0.44 0.33	0.42 0.32	28/37	11/16
	1k	Six-way Composite	0.38 0.42	0.35 0.37	0.37 0.37	0.36 0.37	23/32 14/32	7/16
3 Labels	Full	Four-way Composite	0.60 0.48	0.49 0.40	0.50 0.40	0.50 0.40	26/39	13/15
	1k	Four-way Composite	0.58 0.60	0.45 0.46	0.49 0.48	0.47 0.47	31/39 25/39	13/16

7.3.2 Methods, Assumptions, and Procedures

We developed a Spanish sentiment detection system for Twitter as part of Task 1 of TASS1 2015 (Deas et al., 2015). TASS is an annual workshop focusing on sentiment analysis in Spanish, especially of short social media texts such as tweets. Each year, TASS proposes a number of tasks and collects the results of different participating systems. In 2015, Task 1 was a combined subjectivity/polarity task: for each tweet, the competing system is expected to provide a label. TASS distributes a standard data set of over 68, 000 Spanish tweets for participants in this task.

Instead of creating a new Spanish-specific system, we adapted our existing English system to the Spanish language.

7.3.3 Results and Discussion

Our results in TASS are shown in Figure 23. In this figure all results are shown at the global level, including positive and negative and all measures are macro-averaged.

7.3.4 Conclusions

We showed that with relatively small engineering efforts and the proper resources, but without any language-specific feature engineering, our system can be adapted to a new language and achieve performance that is competitive with other systems at TASS. As a side effect, we formalized the process of adapting our system to any new language.

7.4 Sentiment Analysis using Word Embeddings

7.4.1 Introduction

Neural networks are one of the most popular approaches for many natural language processing tasks such as sentiment analysis. They often outperform traditional machine learning models and achieve the state-of-art results on most tasks. However, many existing deep learning models are complex, difficult to train, and provide limited improvement over simpler methods.

7.4.2 Methods, Assumptions, and Procedures

We developed a simple, robust and powerful model for sentiment classification (Yu et al., 2017). We show that the performance of a linear bag-of-words model increases by naively combining ngrams and word embeddings. We obtain additional improvements by grouping word embeddings according to part-of-speech (POS) tags. We also experimented with other complex ways of incorporating word embeddings that do not improve performance.

Table 24: Results Compared to other Models. The bold and underlined, and bold-only numbers are the best and second best results respectively. Our NBLR + POSwemb performs the best or second on several datasets.

Data	MR	CR	MPQA	Subj	TTwt	IMDB
NBSVM (Wang and Manning, 2012)	79.4	81.8	86.3	93.2	65.6	91.2
NBLR + POSwemb	81.6	84.0	89.9	93.3	<u>69.9</u>	91.8
Li et al., 2016	79.5	81.1	82.1	92.8	–	<u>93.0</u>
DSCNN(Zhang et al., 2016)	81.5	–	–	93.2	–	90.2
BLSTM-2DPooling(Zhou et al., 2016)	81.5	–	–	93.7	–	–
SA-LSTM (Dai and Le, 2015)	80.7	–	–	–	–	92.8
AdaSent (Zhao et al., 2015)	<u>83.1</u>	<u>86.3</u>	<u>93.3</u>	<u>95.5</u>	–	–
CNN non-static (Kim, 2014)	81.2	84.0	89.6	93.4	–	–
bowwvSVM (Zhang and Wallace, 2016)	79.67	81.3	89.7	91.7	–	–
SVM-dep (Jiang et al., 2011)	–	–	–	–	63.3	–
AdaRNN comb (Dong et al., 2014)	–	–	–	–	65.9	–
Targ-dept+(Vo Zhang et al., 2015)	–	–	–	–	<u>69.9</u>	–
TC-LSTM (Tang et al., 2015)	–	–	–	–	69.5	–

7.4.3 Results and Discussion

Results for this work are shown in Figure 24. Our model is called **NLBR+POSwemb**. We see that it has state of the art results on most datasets.

7.4.4 Conclusions

This model outperforms many deep learning models and achieves comparable results to other deep learning models with complex architectures on sentiment analysis datasets.

Chapter 8

Source and Target Belief and Sentiment

8.1 Summary

In this chapter, we present our systems which participated in the 2017 TAC KBP Belief and Sentiment (BeSt) evaluation.

The structure of this chapter is as follows. We first present a shared introduction. We then list our different systems, and for each we have separate discussions of “Methods, Assumptions and Procedures” and of “Results and Discussions”. For some of our system we also present specific conclusions. We then finish the chapter with a conclusion for the whole chapter.

8.2 Introduction

As in 2016, The 2017 TAC KBP BeSt evaluation of source-and-target belief and sentiment covers 24 different scenarios: two phenomena to detect, three languages, two genres, and two conditions. As a result, for the 2017 evaluation (as in 2016), we did not build a unified system; instead, we use several different base approaches which we tailor to different situations. We list them here:

- A sentiment system based on identifying the target only, with the source implicitly the author. This is developed for English (Section 8.3), and adapted for Chinese (Section 8.5) and Spanish (Section 8.6).
- A sentiment system based on a relation extraction system, with the notion that sentiment from source to target is a relation from source to target. This is used only for English (Section 8.4).
- A belief system that combines high-precision word tagging with a high-recall default system. This is used for English (Section 8.7) and adapted for Chinese (Section 8.8).
- A belief system that is based on a weighted random choice of tags, which we use for Spanish (Section 8.9).

While we perform experiments on the two genres (discussion forums and newswire) to find optimized training data for each test data, we do not perform special adaptation for the two conditions of gold and predicted ERE (entities, relations, events). Instead, all results are obtained by training on gold ERE.

Our systems are improved versions of our 2016 systems.

8.2.1 Data

We use the following data sets from the LDC.

- LDC2016E27_DEFT_English_Belief_and_Sentiment_Annotation_V2
- LDC2016E61_DEFT_Chinese_Belief_and_Sentiment_Annotation

- LDC2016E62_DEFT_Spanish_Belief_and_Sentiment_Annotation

We did not use any other sentiment-annotated resources (other than sentiment lexicons). For English and Chinese belief (Section 8.7 and 8.8), we used belief word taggers which are trained on data using a different annotation, in which only targets are identified (the source being always the author):

- LDC2014E55_DEFT_Committed_Belief_Annotation_R1_V1.1 and LDC2014E106_DEFT_Committed_Belief_Annotation_R2
- LDC2015E99_DEFT_Chinese_Committed_Belief_Annotation

For the sentiment systems, on the “Belief and Sentiment” data sets, we used the above data as training, and the BeSt 2016 evaluation data as development data. In this paper, we report results on this dev set for English, as well as for the eval set.

We did not train separately on predicted ERE.

For the baseline, we determine on the training set what the majority value is; for sentiment it is always (across languages and genres) “neg”, for belief always (across languages and genres) “CB”. We then create a sentiment for each possible target: for beliefs, for each relation and each event; for sentiment, for each entity, relation, and event. (We take gold or predicted ERE files, as the case may be, as the source of the EREs.) For the source, we always assume it is the author, so we determine the author and choose the appropriate mention as the source mention. Some newswire files have no author mention, so we fill in “None” for the source (which is what the gold expects). In our belief systems, we use this baseline as one of our systems and as a component in another system; see Section 8.7, Section 8.8, and Section 8.9. We subsequently improved the baseline by extending the set of identified targets by identifying relations and events not only through the trigger, but also through the arguments. This extended baseline is the official baseline for the evaluation, and it outperforms the baseline used in our belief experiments.

All results are given using the “one-is-enough” option for provenance scoring, which requires at least one overlap between the predicted provenance list and the gold provenance list for each predicted sentiment or belief.

8.3 English Sentiment 1

8.3.1 Methods, Assumptions, and Procedures

Basic Approach

We employ widely used text classification features such as word embeddings and sentiment words count. Also, we implement some task-specific features such as the mention types of the target. The features are extracted on the target, sentence, post and file levels. As classifiers, we use Support Vector Machines (SVM) with linear kernels and Random Forest classifiers are used to train our models. We apply two different approaches, target-oriented and context-oriented, with similar features on this task.

Data Pre-Processing

Pre-processing steps include sentence and word tokenization and other common NLP tasks such as part-of-speech tagging and parsing. For the tokenization, we make use of the NLTK package (Bird et al., 2009). Sentences are also syntactically parsed and marked with part-of-speech tags with the Stanford CoreNLP tool (Manning et al., 2014b).

We then collect all possible targets from the ERE file (gold or predicted). We exclude all entity mentions which refer to the author of that particular passage (since the author is unlikely to express sentiment towards herself); these are typically signaled by texts that are simply *I* or *my* in our data set. For each of the other possible targets, we extract the sentence where it is located and save the indicators of which file, post and author it is from.

Approach

The task we address in this evaluation is source-target dependent sentiment analysis, which means we are interested not only in the sentiment towards a target, but also in the person who has this sentiment. Nonetheless, we still assume for the approach discussed in this section that the source is the author. This is because the vast majority of sentiment cases for both discussion forum and newswire data sets are from the author. We pursue two approaches.

First, we apply a target-oriented approach. The main idea of this method is to have target-specific features. One of most challenging problems of this task is that the sentences in our dataset are complex, and they are relatively long. Also, it is common to have more than 5 potential targets in each sentence. The most important step behind our approach is to get a “small sentence” which is most related to the target. In order to get this small sentence, we use the ‘S’ labels from the parsing output to divide the sentence into chunks of clauses and non-clausal words. We then scan the two chunks both before and after the chunk where the target is located, and discard the chunks that contain other mentions and that have the ‘S’ label. Finally, we combine the remaining clauses into a small sentence that includes the target.

We generate features based on our small sentence. Sentiment lexicons including the NRC Emotion Lexicon (Mohammad and Turney, 2010), Bing Liu’s Lexicon (Hu and Liu, 2004), and MPQA Subjectivity Lexicon (Wilson et al., 2005) help count the sentiment words in the small sentence. We added the pre-trained word embedding of each word in the small sentence, weighted by POS (as determined by the tagger). The weights are chosen through experiments on the training set. We also experimented with word2vec (Mikolov et al., 2013a) and different Glove (Pennington et al., 2014) word embedding resources. Glove 840B300d gives us the best performance. Other features include the types of possible target entities, relations and events such as ‘GPE’, ‘LOC’, ‘conflict,’ etc., as well as the relative position of the targets in the sentence.

Second, we apply a context-oriented method. We use the same features described for the first approach. However, we no longer cut the sentence into clauses to get the small sentence. Instead, we have the features of original sentence, post and file.

We apply linear SVMs and Random Forest classifiers in this task for both approaches. Hyper-parameters were tuned via cross-validation.

We experimented with many other features such as different sentiment lexicons, negation, punctuation and POS tagger counts, but they did not prove helpful for this task.

8.3.2 Results and Discussion

The model trained with features of the small sentence gets 6% higher than of the original large sentence on F score if we only work on the sentence level. However, the context-oriented approach outperforms the target-oriented method. We therefore use the context-oriented approach without small sentences.

We reused our results from 2016 concerning genres. Last year, we used a different development set to determine which genres to train on. We found that for DF, the best system to use is trained on DF only, while for NW training on a combination of DF and NW is best.

Results on the evaluation set are shown in Table 26. For the English Sentiment result on the evaluation set, our system using the context-oriented approach performed the best among all of them on all genres and conditions.

8.3.3 Conclusions

We find that the sentiment ratio across different files and genres differs drastically. It would be useful to determine the sentiment level in each file first and apply different approaches to detect sentiment for files with high or low sentiment level.

Table 25: Results for our English Sentiment System-2 on “SuperDev” Data

Train on \ Test on	Disc. Forums			Newswire		
	Prec.	Rec.	F-meas.	Prec.	Rec.	F-meas.
Disc. Forums	35.5%	59.2%	44.4%	7.0%	13.0%	9.9%
Disc. Forums + Newswire	34.5%	57.0%	43.0%	4.0%	4.0%	4.0%

8.4 English Sentiment 2

8.4.1 Methods, Assumptions, and Procedures

Basic Approach

Our general approach is based on relation extraction and directional sentiment as a relation between source and target. Relation extraction typically identifies relations between two entities in raw text. We extend the notion of relation to cover sentiment. The sources of the sentiment relation are always entities, while the target can be entities, events or relations. We use a supervised approach to detect positive or negative sentiment. The English Sentiment System-2 is built on top of the SINNET system, which was developed under DEFT funding and is a particular relation extraction system for social event extraction. Social events are the building blocks of social networks (Agarwal and Rambow, 2010b).

8.4.2 Data Pre-Processing

Some significant pre-processing work was done to adapt the SINNET system to the task of source-and-target sentiment extraction (Figure 9).

We model source-and-target sentiment as a relation between the source and target. However the original system requires both parties to the relation to be mentioned explicitly in the text, and thus does not capture the implicit sentiment of the author. It only considers the sentiment relations between entities within the sentence. In order to capture those cases, we add at the beginning of each sentence the words *The author says*, with *the author* a mention of the author entity (which is already defined in discussion forums).

8.4.3 Approach

The English Sentiment System-2 is based on the SINNET system and relies on it for all machine learning aspects. The SINNET system uses tree kernels and Support Vector Machines (SVMs) to extract social events from natural language text. The SINNET system uses a combination of structures derived from the linear order, syntactic parses (phrase structure trees and dependency trees), and semantic parses. The English Sentiment System-2 takes as training data sentences with pairs of (entity, ERE) marked, and with an annotation for that sentence and (source, target) pair showing what type of sentiment the source has towards the target. The input to the system consists of two types of files: a raw text as well as text with entity and sentiment annotations. There are two main modules. The first is the linguistic pre-processing module which generates all linguistic information that is used by the machine learning (ML) module, and the ML module itself which uses SVMs with tree kernels.

8.4.4 Results and Discussion

For developing our system we conducted a set of experiments on “SuperDev” to determine which model to use on each of the genres, as we did for our other sentiment system (see Section 8.3.2). The results from the experiments are reported in Table 25. As we can see, for both DF and NW, the best system to use is trained on DF only.

Table 26: Results for our two English Sentiment systems on KBP Eval Data, along with the baseline

System	Genre	Gold ERE			Predicted ERE		
		Prec.	Rec.	F-meas.	Prec.	Rec.	F-meas.
Baseline	Disc. Forums	12.2%	61.3%	20.4%	5.6%	9.6%	7.1%
	Newswire	7.3%	35.8%	12.2%	2.1%	3.4%	2.6%
System 1	Disc. Forums	19.5%	35.1%	25.0%	9.7%	6.5%	7.8%
	Newswire	6.7%	28.2%	10.8%	2.7%	9.0%	4.2%
System 2	Disc. Forums	11.5%	47.0%	18.5%	N/A	N/A	N/A
	Newswire	8.0%	10.1%	8.9%	N/A	N/A	N/A

Results on the evaluation set are shown in Table 26. We see that our system 2 performs slightly worse than our system 1.

8.4.5 Conclusions

We plan to run experiments with different linguistic features, in particular semantic. We also plan to add sentiment-relevant features such as lexical features derived from the sentiment lexicons (which we have not yet used in our system 2). We plan to also work on those cases in which the source is not the author, since our system 2 based on SINNET can find those cases, while our system 1 cannot.

The training data is very skewed. There are many more pairs of entities without sentiment than pairs of entities that have sentiment. We can use random under-sampling and random over-sampling techniques to achieve better results. Also, the current system needs to be boosted with sentiment features in order to improve its accuracy.

8.5 Chinese Sentiment

8.5.1 Methods, Assumptions, and Procedures

Basic Approach

We follow the same approach as for English sentiment 1 (Section 8.3).

Data Pre-Processing

We segment sentences based on a list of some specific punctuations. For Chinese, it is critical to group words into meaningful phrases and to do POS tagging. This is performed by the Jiaba text segmentation package¹. In addition, we treat all mentions except the author mentions as possible targets.

Approach

We apply the context-oriented method described in the English Sentiment 1 system (Section 8.3) to the Chinese sentiment task. We used Polyglot Chinese (Al-Rfou et al., 2013) to obtain word embeddings, and the HowNet Chinese Sentiment Lexicon².

8.5.2 Results and Discussion

Because there are many annotation errors and few sentiment cases in the Chinese dataset, the model trained on this data does not perform well.

For developing our system we conducted a set of experiments on our dev set to determine which model to use. The results from the experiments are reported in Table 27. Also, our results on the real evaluation set are shown in Table 28.

¹<https://github.com/fxsjy/jieba>

²http://www.keenage.com/html/e_index.html

Table 27: Results for our Chinese Sentiment System on the development dataset

Train on \ Test on	Disc. Forums		
	Prec.	Rec.	F-meas.
Disc. Forums	26.3%	6.8%	10.8%

Table 28: Results for our Chinese Sentiment system on KBP Eval Data, along with the baseline

System	Genre	Gold ERE			Predicted ERE		
		Prec.	Rec.	F-meas.	Prec.	Rec.	F-meas.
Baseline	Disc. Forums	4.1%	49.7%	7.6%	2.6%	9.1%	4.0%
	Newswire	1.9%	47.8%	3.7%	0.5%	4.5%	0.9%
System 1	Disc. Forums	19.4%	1.8%	3.4%	0.0%	0.0%	0.0%
	Newswire	27.3%	1.3%	2.6%	0.0%	0.0%	0.0%

There were a total of 7 systems submitted for all genres and conditions of Chinese sentiment. Our systems didn't do well on the Chinese Sentiment task.

8.5.3 Conclusions

Because there are many annotation errors and few sentiment cases in the Chinese dataset, the model trained on this data performs badly. It would be helpful to have a new and similarly labeled dataset, or to investigate the use of alternate annotations.

8.6 Spanish Sentiment

8.6.1 Methods, Assumptions, and Procedures

Basic Approach

The main contributions of this approach involve generalizations of English capabilities for Spanish sentiment between a source and a target. We use a pre-processing step that limits the target range to only a relevant span of words. We also trained classifiers using features based on both Spanish lexicons and word embeddings. The pre-processing and training steps require Spanish dictionaries, corpora, tokenizers, and parsers. Reuse of these methods for other languages and domains would require similar resources.

Data Pre-Processing

Similar to English pre-processing, we tokenize and parse the data to determine the relevant text. For Spanish, we used the Stanford CoreNLP tokenizer, POS tagger, and parser (Manning et al., 2014b). The relevant Spanish phrasal categories to determine the small sentences are similar to English- 'S' also marks a clause.

Approach

The overall approach for Spanish is to use word embeddings combined with lexical features and other document features. We first identify the words in the relevant range for the target. Then we identify the word embedding for each word in this range and average these word embeddings to obtain a representation for the entire sequence. The word embeddings are the 300-dimensional embeddings created from the Spanish Billion-Word Corpus (Cardellino, 2016). We use a Spanish lexicon to count the number of polar words in each data instance (Perez-Rosas et al., 2012). We also included features to account for the overall polarity in a file, in an individual post, or from a specific author. These features were represented as an average of the

Table 29: Results for our two Spanish Sentiment systems on TAC KBP Evaluation Data

System	Genre	Gold ERE			Predicted ERE		
		Prec.	Rec.	F-meas.	Prec.	Rec.	F-meas.
Baseline	Disc. Forums	9.7%	65.2%	16.8%	5.2%	9.5%	6.7%
	Newswire	4.4%	37.8%	7.9%	2.3%	5.6%	3.2%
<i>System1</i>	Disc. Forums	12.6%	58.0%	20.7%	6.6%	7.7%	7.1%
	Newswire	6.8%	32.5%	11.2%	2.9%	3.8%	3.3%

target range embeddings for all data in a file, post, or from the same author. Finally, we included a feature based on the mention type of the target. The idea is that certain categories of mentions (i.e., businesses, life events) are more or less inherently likely to include sentiment towards the respective target. This feature was represented as a categorical, one-hot encoded feature.

We experimented with different training variations. For instance, in one variation we allowed the word embeddings to vary rather than be fixed for the entirety of training. Another variation is to learn a weighted average of the embeddings for each target range using an attention mechanism over words. We also experimented with averaging embeddings only over the content words (i.e., after removal of Spanish stop-words.). In addition, rather than a categorical representation of the mention type feature, we created an embedding for the mention type (varying between 10 and 15 dimensions). We experimented with linear SVMs. Hyperparameters were tuned via cross-validation.

8.6.2 Results and Discussion

In Table 29, we report the results on TAC KBP evaluation data using a linear SVM (*System1*) where the features included the average of the words in the target range, post, author, and file as well as a polarity score and the mention type as a categorical feature.

Two teams submitted a total of 4 systems. *System1* was the best performing system on all genres and conditions.

8.6.3 Conclusions

Our ongoing and future work include utilizing novel kernels for word embeddings, such as the model introduced in (Ghosh et al., 2015b) where the authors utilized a greedy alignment technique to measure the similarity between two data instances as kernel similarity. Also, although we have utilized some lexicons in our current experiments, we are interested in investigating more lexicons, for instance the Spanish LIWC (Pennebaker et al., 2001).

8.7 English Belief

8.7.1 Methods, Assumptions, and Procedures

Approach

Our approach is to use an existing English word-level belief tagger (Prabhakaran et al., 2010a; Werner et al., 2015a). It tags each word which is the syntactic head of a proposition with one of the belief labels:

- CB for committed belief where the writer strongly believes in the proposition.
- NCB for non-committed belief, where the writer has a weak belief in the proposition.
- ROB for reported belief, which is the case when the writer’s intention is to report on someone else’s stated belief, whether or not they themselves believe it or not.

Table 30: Results for our three English Belief systems on KBP Eval Data, along with the official baseline

System	Genre	Gold ERE			Predicted ERE		
		Prec.	Rec.	F-meas.	Prec.	Rec.	F-meas.
System 1	Disc. Forums	78.86%	74.40%	76.57%	3.23%	0.33%	0.60%
	Newswire	77.82%	52.10%	62.41%	2.86%	0.43%	0.75%
System 2	Disc. Forums	84.11%	27.90%	41.90%	0.00%	0.00%	0.00%
	Newswire	80.84%	15.51%	26.02%	0.00%	0.00%	0.00%
System 3	Disc. Forums	72.07%	84.19%	77.66%	2.76%	0.35%	0.62%
	Newswire	77.41%	53.79%	63.48%	2.81%	0.45%	0.77%

- NA for propositions which are not beliefs. NA is discarded as it is not used in this evaluation.

All other words are tagged as “Other”. We then check whether the trigger for a relation mention or an event mention contains a word that has been tagged with a belief tag, and then apply that tag to the entire relation or event mention. We combine this approach with a majority baseline for our best results.

We have three systems (note that we do not represent them in order of their submission number, but in an order that allows an easier exposition):

- The word-tagger based system (System 2). In this system three sets of files have been applied in order to generate the final output, including input text files, output of the English belief tagger, and the ERE files (gold or predicted, as the case may be). When extracting target entities and relations from the ERE files we look for the associated committed belief tags within the output of the English belief tagger. Moreover, using the word offset we find the corresponding authors for each case. This is a high-precision, low-recall system.
- The majority-baseline system (System 3). The majority Baseline labels all the extracted target entities, and relations as the majority belief tag, which is ‘CB’. The source is again the author of the passage which contains the target mention. This is a high-recall, lower-precision system.
- The combination system (System 1). System 1 employs both systems within the single pipeline. We applied System-3 to label all the untagged entities and relations, which were not tagged by the committed belief tagger and tag them as ‘CB’. Applying the combination system could lead to significant improvements in our experiments.

8.7.2 Results and Discussion

The results on the evaluation set are shown in Table 30. Our System 3 is a majority baseline. On discussion forum (which have more varied types of beliefs compared to newswire), the precision of both combination System 1 and System 2 outperforms the majority baseline (System3). In contrast, for both newswire and discussion forums, the baselines have a good recall, so that our combined System 1 could not beat the baseline in recall.

In comparison to the other submissions, for newswire with gold ERE, we are the best performing system, while in discussion forums another system beat our System3 by a small margin. In predicted ERE the majority baseline outperforms all the systems.

8.7.3 Conclusions

We will start out by incorporating the improved baseline system into our belief prediction system. We are in the progress of adding a new set of features such as word embeddings, sentiment, negation and hedge words.

In addition to novel features, various machine learning approaches will be applied in order to improve the overall performance of the Belief tagger.

We will also experiment using the approach based on relation extraction that we used for sentiment in our second system (Section 8.4). This system can detect sources which are not the author, and we expect it to contribute to newswire, where cases of reported belief are more common.

8.8 Chinese Belief

8.8.1 Methods, Assumptions, and Procedures

Basic Approach

Our basic approach is similar to the approach we use for English (Section 8.7). We use an existing Chinese word-level belief tagger. Like the English word-level belief tagger, it tags each word which is the syntactic head of a proposition with one of the belief labels. All other words are tagged as “Other”. We then check whether the trigger for a relation mention or an event mention contains a word that has been tagged with a belief tag, and then apply that tag to the relation or event mention.

Data Pre-Processing

Because Chinese does not have space between words, we first use the Stanford Chinese word segmenter (Manning et al., 2014b) to split sentences into sequences of words.

Approach

As in English, we have three systems:

- The word-tagger based system (System 2). We use the belief word-tagger presented in (Colomer et al., 2016a). We part-of-speech tag the text. Then for each word, the system extracts the word, three words before this word, one word after this word, and their pos tags as a features. We predict the belief tag.
- The majority-baseline system (System 3).
- The combination system (System 1).

8.8.2 Results and Discussion

The results on the evaluation set are shown in Table 31.

8.8.3 Conclusions

Our findings are as follows.

- For predicted ERE, no system, including the baseline, found any beliefs, since no relations and few events were predicted; therefore, we do not show the predicted ERE condition. Since all the scores for Predicted ERE are 0, we are only going to compare the result of Gold ERE with DF and NW.
- We first look at discussion forums with gold ERE. As expected, our word-tagger System 2 has very low recall but a high precision. Combining our baseline system (System 3) with System 2 yields the system with the higher precision, but at the cost of much lower recall in comparison to the baseline. Thus, in terms of f-measure, our best system is, disappointingly, our baseline system, System 3.
- We now consider newswire with gold ERE. Among our three systems, our word-tagger System 2 found the lowest number of beliefs, and thus combination of System 2 with our baseline System 3 does not significantly improve System 3.
- For the predicted ERE, no system predicts any belief, and we omit the numbers.

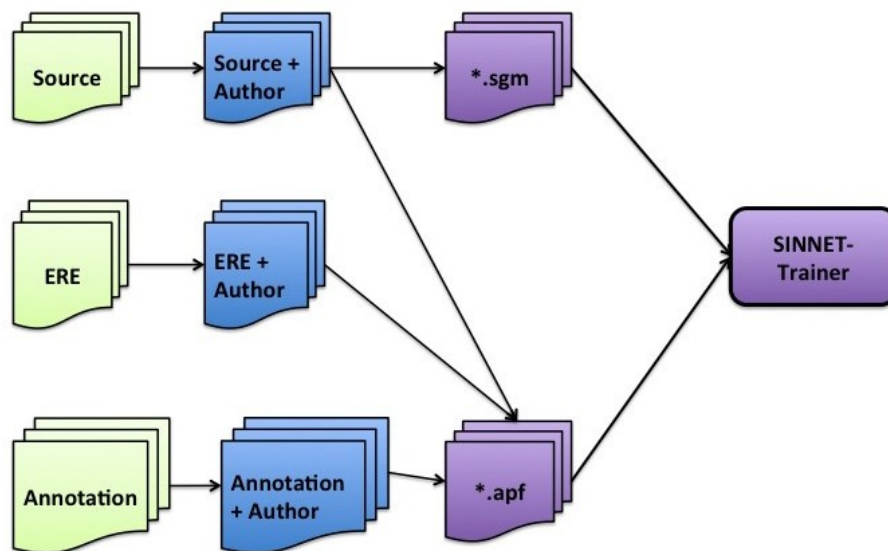


Figure 9: A pipeline to add “The author says”; convert ERE-annotated files into the proper APF (Ace Program Format) files and convert Source files to *.sgm format which is the format expected by the SINNET system.

Table 31: Results for our three Chinese Belief systems on KBP Eval Data, along with the official baseline. Note that no belief was predicted on the predicted ERE, neither by the official baseline, nor by our systems; we therefore omit this condition from the table.

System	Genre	Gold ERE		
		Prec.	Rec.	F-meas.
System 1	Disc. Forums	81.56%	69.41%	74.99%
	Newswire	80.30%	47.89%	60.00%
System 2	Disc. Forums	78.73%	25.69%	38.74%
	Newswire	78.47%	19.05%	30.65%
System 3	Disc. Forums	80.67%	82.15%	81.40%
	Newswire	81.50%	51.98%	63.48%

Table 32: Results for our three Spanish Belief systems on KBP Eval Data, along with the official baseline. Note that no belief was predicted on the predicted ERE, neither by the official baseline, nor by our systems; we therefore omit this condition from the table.

System	Genre	Gold ERE		
		Prec.	Rec.	F-meas.
System 1	Disc. Forums	65.30%	70.64%	67.87%
	Newswire	72.14%	56.30%	63.25%
System 3	Disc. Forums	78.68%	75.68%	77.15%
	Newswire	79.95%	59.47%	68.20%

8.9 Spanish Belief

8.9.1 Methods, Assumptions, and Procedures

For Spanish belief, we adopted a simpler approach, which we tuned on the entire data set.

- System 1 is a baseline system, with a strict matching against targets.
- System 3 is a slight variant of System 1 (an attempt at finding more targets), which did not produce the desired results.

8.9.2 Results and Discussion

The results on the evaluation set are shown in Table 32. For predicted ERE, no system, including the baseline, found significant number of beliefs, since no relations and few events were predicted; therefore, we do not show the predicted ERE condition. Our System 3 beat the other systems submitted on both genres.

8.10 Conclusion for All Work in This Chapter

There are several clear areas for us to improve our existing systems.

1. We need to train systems on predicted ERE data.
2. We performed extensive experiments with our relation-extraction system (Section 8.4) in order to add sentiment-specific features. These did not improve the performance of the system. However, we obtained good results on belief on newswire, but the results were too late for the evaluation and this paper. We will report them in the future.
3. We need to investigate in more detail how to train systems for different conditions. Specifically, even within a genre such as discussion forums, the amount of sentiment expressed can differ from what was encountered in training, which can decrease results.

We expect the performance of automatic source-and-target belief-and-sentiment systems to greatly increase over the next few years as the data is better understood and appropriate features and machine learning techniques are discovered.

Chapter 9

Identifying Hedging

9.1 Summary

We have developed a novel method of identifying *hedge* terms using a set of manually constructed rules. We performed experiments adding hedge features to our English committed belief system to improve classification. We compared performance of this system (a) without hedging features, (b) with dictionary-based features, and (c) with rule-based features. We found that using hedge features improves performance of the committed belief system, particularly in identifying instances of *non-committed belief* and *reported belief*.

9.2 Introduction

Hedging refers to the use of words, sounds, or constructions that add ambiguity or uncertainty to spoken or written language. Hedges are often used by a speaker to indicate lack of commitment to what they say. So, the ability to accurately classify words and phrases as hedges is very relevant to the task of committed belief tagging—that is, determining the level of commitment a speaker has toward the belief expressed in a given proposition. A major challenge in identifying hedges is that many hedge words and phrases are ambiguous.

Our goal was to develop a new method of disambiguating potential hedges using a set of manually constructed rules and to use this method to improve the performance of our English committed belief tagger.

9.3 Methods, Assumptions, and Procedures

Table 33: List of (potential) hedge words and phrases

Relational Hedges	Propositional Hedges
according to , appear , arguably, assume , believe , consider , could, doubt , estimate, expect, feel , find , guess, hear, I mean, I would say, imagine , impression , in my mind, in my opinion, in my understanding, in my view, know , likely , look like, looks like, may , maybe, might , my thinking, my understanding, necessarily, perhaps, possibly, presumably, probably, read, say, seem, seemingly, should , sound like, sounds like, speculate, suggest, suppose , sure , tend , think, understand, unlikely , unsure	a bit, a bunch, a couple, a few, a little, a whole bunch, about , allegedly, among others, and all that, and so forth, and so on, and suchlike, apparently, approximately, around, at least, basic, basically, completely , et cetera, etc, fair , fairly , for the most part, frequently, general , generally, in a way, in part, in some ways, kind of, kinda, largely, like , mainly, more or less, most , mostly, much, occasionally, often , partial , partially, partly, possible , practically, pretty , pretty much, probable, rarely, rather , really , relatively, rough, roughly , seldom, several, something or other, sort of, to a certain extent, to some extent, totally , usually, virtually

9.3.1 Identifying Hedge Terms

We have compiled a dictionary of 117 potential hedge words and phrases, divided into *relational* and *propositional* hedges. As described in Prokofieva and Hirschberg (2014), relational hedges have to do with the speaker’s relation to the propositional content, while propositional hedges are those that introduce uncertainty into the propositional content itself. Table 33 show more examples of these two types of hedges.

Our baseline hedge detector is a simple, *dictionary-based* one. Using our dictionary of potential hedge terms, we look up the lemma of each token in the dictionary and mark it as a hedge if found. This procedure, however, does not take into account the inherent ambiguity of many of the hedge terms. To handle this ambiguity, we implemented *rule-based* hedge detection. The rule-based system disambiguates hedge vs. non-hedge usages using rules based on context, part-of-speech, and dependency information.

The full list of hedge words and phrases in our dictionary is shown in Table 33. The hedge terms for which we have written rules are shown in bold; the rule-based system classifies others as hedges by default. Table 34 shows a sample of the rules, with examples of hedging and non-hedging uses. We evaluate both

Table 34: Examples of rules used to disambiguate hedge terms

Hedge term	Rule	Examples
about	If token t has part-of-speech IN, t is non-hedge. Otherwise, hedge.	Hedge: There are about 10 million packages in transit right now. Non-hedge: We need to talk about Mark.
rather	If token t is followed by token ‘than’, t is non-hedge. Otherwise, hedge.	Hedge: She’s been behaving rather strangely. Non-hedge: She seemed in-different rather than angry.
assume	If token t has ccomp dependent, t is hedge. Otherwise, non-hedge.	Hedge: I assume his train was late. Non-hedge: When will the president assume of-fice?
tend	If token t has xcomp dependent, t is hedge. Otherwise, non-hedge.	Hedge: Written language tends to be formal. Non-hedge: Viola tended plants on the roof.
appear	If token t has xcomp or ccomp dependent, t is hedge. Otherwise, non-hedge.	Hedge: The problem appears to be a bug in the software. Non-hedge: A man suddenly appeared in the doorway.
sure	If token t has neg dependent, t is hedge. Otherwise, non-hedge.	Hedge: I’m not sure what the exact numbers are. Non-hedge: He is sure she will turn up tomorrow.
completely	If the head of token t has neg dependent, t is hedge. Otherwise, non-hedge.	Hedge: That isn’t completely true. Non-hedge: I am completely sure you will win.
suppose	If token t has xcomp dependent d and d has mark dependent ‘to’, t is non-hedge. Otherwise, hedge.	Hedge: I suppose the package will arrive next week. Non-hedge: I’m supposed to call if I’m going to be late.
should	If token t has relation aux with its head h and h has dependent ‘have’, t is non-hedge. Otherwise, hedge.	Hedge: It should be rainy tomorrow. Non-hedge: He should have been more careful.

dictionary-based and rule-based approaches in a committed belief tagger.

9.3.2 Hedge Features

For the experiments described in this paper, we add the following additional features to the committed belief tagger:

- **Word features:** based on properties of the current word being tagged. If the word is classified as a hedge by the hedge detector, *HedgeLemma*, and *HedgeType* are set to the token, lemma, and hedge type (propositional or relational) of the word. Otherwise, these features are null.
- **Dependency features:** based on attributes of words related to the current word by the dependency parse. If the child of a given word is classified as a hedge by the hedge detector, *HedgeTokenChild*, *HedgeLemmaChild*, and *HedgeTypeChild* are set to the token, lemma, and hedge type (propositional or relational) of the child. Otherwise, these features are null. Likewise, we define *HedgeToken*{Parent,Sibling,DepAncestor}, *HedgeLemma*{Parent,Sibling,DepAncestor}, and *HedgeType*{Parent,Sibling,DepAncestor} if the parent, sibling, or ancestor of the word is classified as a hedge.
- **Sentence features:** based on properties of the sentence containing the current word. If the hedge detector identifies a hedge anywhere in the sentence, *SentenceContainsHedge* is set to true.

9.4 Results and Discussion

3. I'm always kind of amused that you guys want to fire all the government workers because they are making too much money, but you have a fit when someone undercuts your salary.
Is the meaning of the word <i>kind of</i> closer to:
☐ • type of ("This specimen is a <u>kind of</u> berry as indicated by the seeds located on its skin.")
☐ • to some extent ("It's <u>kind of</u> hard to read them straight up and down like that.")

Figure 10: Example of AMT word disambiguation task

9.4.1 Experiments and Results

All the experiments reported below use 5-fold cross validation on the 2014 Darpa DEFT Committed Belief Corpus (Release No. LDC2014E55). The documents in this corpus are from English discussion forum data. We compare the performance of the system using (a) no hedge features (b) hedge features obtained using the dictionary-based tagger, and (c) hedge features obtained using the rule-based tagger. Results are shown in Table 35.

As we might expect, hedge features are most significant in detecting instances of reported belief and non-committed belief. Since these represent only a small portion of the full corpus, the effect on the overall performance is not large. However, it is still significant. Using dictionary-based hedge features, we see an increase of 1.82 in the f-measure for ROB as compared to the baseline, from 23.29 to 25.11, and an increase of 2.29 for NCB, from 23.66 to 25.95. The overall f-score increases 0.43, from 67.52 to 69.95. Using rule-based hedge features, the increase compared to the baseline is more significant. For ROB, the f-score shows an increase of 4.14, from 23.29 to 27.43. For NCB, the f-score increases 6.77, from 23.66 to 30.43. The overall increase in the f-score using the rule-based hedge features is 0.55, from 67.52 to 68.07.

Table 35: Belief results using (a) no hedge detection, (b) dictionary-based hedge detection, and (c) rule-based hedge detection.

Tag (count)	Precision	Recall	F-measure
ROB (256)	28.02	19.92	23.29
NCB (193)	44.93	16.06	23.66
NA (2762)	77.49	56.34	65.24
CB (4299)	69.80	74.78	72.21
Overall	70.69	64.62	67.52

(a)

Tag (count)	Precision	Recall	F-measure
ROB (256)	30.22	21.48	25.11
NCB (193)	49.28	17.62	25.95
NA (2762)	77.69	56.73	65.58
CB (4299)	70.27	75.04	72.58
Overall	71.18	65.01	67.95

(b)

Tag (count)	Precision	Recall	F-measure
ROB (256)	31.63	24.22	27.43
NCB (193)	50.60	21.76	30.43
NA (2762)	77.89	56.52	65.51
CB (4299)	70.58	74.95	72.70
Overall	71.36	65.07	68.07

(c)

9.5 Conclusions

We have shown that hedge detection can improve the performance of a committed belief tagger, particularly in identifying instances of reported belief and non-committed belief. Using hedge features based on simple dictionary-lookup improves performance compared to the baseline; the addition of manually constructed rules improves performance further. While these results are promising, there are limits to the rule-based approach we have used. In many cases, it is not straightforward to define a simple rule disambiguating hedge from non-hedge use. To address these issues, we have used Amazon Mechanical Turk to construct a corpus of forum posts labeled with hedge information. Although other labeled corpora exist, these are in other domains and may not apply to the forum data we are using. After finding potential hedges in the forum posts from the 2014 DEFT Committed Belief Corpora (Release No. LDC2014E55, LDC2014E106, and LDC2014E125), we have presented each potential hedge to turkers as a highlighted word or phrase within a sentence. Rather than asking turkers to label the word as a hedge or not, we showed them definitions and examples of the hedging and non-hedging uses of the term and asked workers which most closely matches the meaning of the word. In future work, we will use this corpus to evaluate the rule-based hedge detector and to train machine learning classifiers directly from the labeled corpus. By this means, we hope to continue to improve the performance of the committed belief tagger as well.

Chapter 10

Sarcasm Detection

10.1 Summary

Recognizing sarcasm is important for understanding people’s actual sentiments and beliefs. For example, failing to recognize the following message as being sarcastic “I love that I have to go back to the emergency room”, will lead a sentiment and opinion analysis system to infer that the author’s sentiment is positive towards the event of “going to the emergency room”. We have developed algorithms for detecting sarcasm in social media posts (both Twitter and Discussion forums) both when considering the utterance in isolation (Ghosh et al., 2015a; Muresan et al., 2016) and when considering the conversation context (Ghosh et al., 2017). Most of our work has focused on English, but we have developed algorithms for Spanish as well. Some of the algorithms were integrated in BBN’s Adept framework. Adept is a scalable, multi-layered, plug-and-play framework developed by BBN and others under the DARPA DEFT program. Adept enables the rapid integration of natural language processing (NLP) algorithms. Our work has led to two awards: the 3rd prize at the NYC-Medial Lab Summit (Ghosh et al., 2015a) and a best paper award at SigDial 2017 (Ghosh et al., 2017). We present below these two specific lines of work.

10.2 Sarcasm Detection as a Word-Sense Disambiguating Problem

10.2.1 Introduction

Sarcasm is generally characterized as a figure of speech that involves the substitution of a literal by a figurative meaning, which is usually the *opposite* of the original literal meaning. We re-frame the sarcasm detection task as a type of word sense disambiguation problem, where the *sense* of a word is either *literal* or *sarcastic*. We call this the Literal/Sarcastic Sense Disambiguation (LSSD) task. We address two issues: 1) how to collect a set of target words that can have either literal or sarcastic meanings depending on context; and 2) given an utterance and a target word, how to automatically detect whether the target word is used in the literal sense (e.g., “I *love* to take a nice stroll in the park every morning”), or in a sarcastic sense (e.g., “I *love* going to the dentist.”). For the latter, we investigate several distributional semantics methods and show that a Support Vector Machine (SVM) classifier with a modified kernel using word embeddings achieves a 7-10% F1 improvement over a strong lexical baseline.

10.2.2 Methods, Assumptions and Procedures

Identify a Set of Target Words

To address the first challenge, we need to identify a set of words from sarcastic utterances, which have a figurative/sarcastic sense (e.g., “love” in the utterance “I love going to the dentist”). We propose a crowdsourcing task where Turkers on the Amazon Mechanical Turk (MTurk) platform are given sarcastic utterances (tweets labeled with #sarcasm or #sarcastic hashtags) and are asked to re-phrase those messages so that they convey the author’s intended meaning (“I *love* going to the dentist” can be rephrased as “I *hate* going to the

dentist” or “I *don’t like* going to the dentist”). Given this parallel dataset, we use unsupervised alignment techniques to identify semantically opposite words (e.g., “love” $\longleftrightarrow$ “hate”, “brilliant” $\longleftrightarrow$ “stupid”, “never” $\longleftrightarrow$ “always”). The words from these pairs that appear in the original sarcastic utterances are then considered as our collection of target words (e.g., “love”, “brilliant”, “never”) that can have both a sarcastic and a literal sense depending on the context. The resulting number of words was 70.

Literal/Sarcastic Sense Disambiguation Task

The task is framed as follow: given an utterance and a target word, automatically detect whether the target word is used in the literal or sarcastic sense. To address this challenge we compare several distributional semantic methods generally used in word sense disambiguation tasks.

Collecting Training/Test datasets. To collect training and test datasets for each of the target words, we use Twitter messages that contain those words. For the *sarcastic sense* (S), we use tweets that contain the target word and are labeled with the #sarcasm or #sarcastic hashtags. For the *literal sense* (L), we collect tweets that contain the target word and are not labeled with the #sarcastic or #sarcasm hashtags. In addition, for the *literal sense*, we also consider a special case, where the tweets are labeled with either positive or negative hashtags (e.g., #happy, #sad). For the 70 target words we identified above, we collected a total of 2,542,249 tweets via Twitter API . We considered a setup where 80% of data is used for training, 10% for development, and 10% for test. We empirically set the number of minimum training instances for each sense of the target word to 400 without any upper restriction. This resulted in 37 target words to be used in the LSSD experiments. The size of training data ranges from 26,802 for the target word “love” to 427 for the word “mature”. As we will see in the results sections, however, the size of the training data is not always the key factor in the LSSD task, especially for the methods that use word embeddings.

Learning Models. We considered two classical approaches used in word sense disambiguation tasks: 1) distributional approaches where each sense of a target word is represented as a context vector derived from the training data; and 2) classification approaches (S vs. L ; S vs. L_{sent}) for each target word. In this report we briefly mentioned the classification approaches. We have two classification tasks: S vs. L and S vs. L_{sent} for each of the 37 target words. We use the libSVM toolkit. Development data is used for tuning parameters. **SVM Baseline:** The SVM baseline for LSSD tasks uses n-grams and lexicon-based binary-valued features that are commonly used in existing state-of-the-art sarcasm detection approaches (González-Ibáñez et al., 2011). They are derived from i) bag-of-words (BoW) representations of words, ii) LIWC dictionary, and iii) a list of interjections (e.g., “ah”, “oh”, “yeah”), punctuations (e.g., “!”, “?”), and emoticons collected from Wikipedia. *CMU Tweet Tokenizer* is employed for tokenization. We kept unigrams unchanged when all the characters are uppercase (e.g., “NEVER” in “A shooting in Oakland? That NEVER happens! #sarcasm”) but otherwise words are converted to lower case. We also change all numbers to a generic number token “22”. To avoid any bias during experiments, we removed the target words from the tweets as well as any hashtag used to determine the sense of the tweet (e.g., #sarcasm, #sarcastic, #happy, #sad). **SVM with MVMEwe Kernel:** We propose a new kernel kernel-we to compute the semantic similarity between two tweets u_r and u_s using the *maximum-valued matrix-element* (MVME) algorithm modified to be able to use word embeddings (MVMEwe). We experimented with this kernel and three types of word embeddings (Weighted Textual Matrix Factorization, word2vec, and GloVe). The similarity measure in the kernel measures the similarity between two tweets u_r and u_s . We augment this kernel kernelWe into libSVM and during evaluation we run supervised LSSD classification for each target word t separately.

10.2.3 Results and Discussion

For brevity, we only report the average Precision (P), Recall (R), and F1 scores with their standard deviation (SD) (given by ‘ $\pm$ ’), and the targets with maximum/minimum F1 scores. W2vSG and W2vCBOW represent

Table 36: Evaluation of classification approaches (SVMbl and kernelWe) for LSSD experiments

Expr.	Senses	Avg. P	Avg. R	Avg. F1	Max. F1(Target)	Min. F1(Target)
SVMbl	S	87.0 $\pm$ 3.3	85.6 $\pm$ 3.1	86.3 $\pm$ 2.7	91.7(yeah)	75.4(sweet)
	L	85.9 $\pm$ 2.8	87.1 $\pm$ 3.6	86.5 $\pm$ 2.8	91.8(yeah)	76.1(sweet)
	L_{sent}	77.3 $\pm$ 4.6 77.8 $\pm$ 3.7	78.2 $\pm$ 4.2 76.7 $\pm$ 6.4	77.7 $\pm$ 3.8 77.1 $\pm$ 4.7	85.5(love) 85.8(love)	68.6(brilliant) 64.6(attractive)
kernelWTMF	S	94.1 $\pm$ 2.2	94.6 $\pm$ 1.8	94.3 $\pm$ 1.8	97.3(brilliant)	88.3(joy)
	L	94.6 $\pm$ 1.8	94.0 $\pm$ 2.3	94.3 $\pm$ 1.9	97.2(mature)	87.9(joy)
	L_{sent}	79.0 $\pm$ 4.6 78.8 $\pm$ 3.7	78.8 $\pm$ 4.4 78.9 $\pm$ 4.9	78.8 $\pm$ 3.8 78.8 $\pm$ 3.6	84.8(mature) 85.4(mature)	61.0(genius) 63.5(genius)
kernelGloVe	S	95.7 $\pm$ 1.6	97.4 $\pm$ 1.7	96.5 $\pm$ 1.1	99.1(mature)	92.9(glad)
	L	97.4 $\pm$ 1.6	95.6 $\pm$ 1.7	96.5 $\pm$ 1.2	99.1(mature)	92.7(interested)
	L_{sent}	79.5 $\pm$ 3.5 82.2 $\pm$ 3.0	83.1 $\pm$ 3.0 78.3 $\pm$ 4.4	81.2 $\pm$ 2.8 80.2 $\pm$ 3.4	86.9(joy) 86.6(joy)	74.2(attractive) 69.2(attractive)
kernelW2vSG	S	96.6 $\pm$ 1.1	98.5 $\pm$ 0.6	97.5 $\pm$ 0.4	99.2(cute)	93.8(interested)
	L	98.5 $\pm$ 0.7	96.5 $\pm$ 1.2	97.5 $\pm$ 0.5	99.2(cute)	93.5(interested)
	L_{sent}	81.9 $\pm$ 3.8 87.0 $\pm$ 3.2	88.1 $\pm$ 3.2 80.2 $\pm$ 4.7	84.8 $\pm$ 3.0 83.4 $\pm$ 3.5	88.8(love) 88.8(love)	74.2(genius) 73.3(genius)
kernelW2vCBOW	S	96.4 $\pm$ 1.0	98.2 $\pm$ 1.1	97.3 $\pm$ 0.6	99.1(mature)	93.8(interested)
	L	98.2 $\pm$ 1.1	96.3 $\pm$ 1.1	97.2 $\pm$ 0.7	99.1(mature)	93.5(interested)
	L_{sent}	81.7 $\pm$ 3.8 87.4 $\pm$ 2.9	88.6 $\pm$ 2.9 79.9 $\pm$ 4.8	84.9 $\pm$ 2.8 83.4 $\pm$ 3.4	89.5(love) 89.2(love)	74.8(genius) 74.4(genius)

the skip-gram and CBOW models implemented in word2vec, respectively.

Table 36 outlines the LSSD experiments using the classification approaches: SVM baseline (SVMbl) and SVM using the kernelWe with word embeddings (kernelWTMF, kernelGloVe, kernelW2vSG, and kernelW2vCBOW). The classification approaches give better performance compared to the distributional approaches. Our new SVM kernel method using word embeddings shows significantly better results when compared to the SVMbl (and distributional approaches). For instance, for the S vs. L task, the average F1 is 96-97%, which is more than 10% higher than SVMbl. Similarly, for S vs. L_{sent} task, F1 scores reported by the kernel using word2vec embeddings are in the range of 83%-84% compared to 77% given by the SVMbl, showing an absolute increase of 7%. As stated earlier, MVME algorithm aligns similar word pairs found in its inputs and this performs well for short texts (i.e., tweets). Thus, the MVME algorithm combined with word embedding in kernelWe results in very high F1. Among the word embedding models, word2vec models give marginally better results compared to GloVe and WTMF, and GloVe outperforms marginally WTMF. The average F1 scores for S vs. L task are higher than the S vs. L_{sent} results.

In terms of the best and worst performing targets, SVMbl prefers targets with more training data (e.g., “yeah”, “love” vs. “sweet”, “attractive”; see Table 36). In contrast, word embedding models for “joy” and “mature”, two targets with comparatively low number of training instances have achieved very high F1 using both distributional and classification approaches (Table 36). This can be explained by the fact that for words, such as “joy”, “mature”, “cute”, and “brilliant”, the contexts of their literal and sarcastic sense are quite different, and word embeddings are able to capture the difference. Overall, out of 37 targets, only 5 targets (“mature”, “joy”, “cute”, “love”, and “yeah”) achieved “maximum” F1 scores in various experimental settings (Table 36) whereas targets such as “interested”, “genius”, and “attractive” achieved low F1 scores. In terms of variance in results, SVM results show low SD (0-4%).

10.2.4 Conclusion

We proposed a reframing of the sarcasm detection task as a type of word sense disambiguation problem, where the sense of a word is its sarcastic or literal sense. Using a crowdsourcing experiment and unsupervised methods for detecting semantically opposite phrases, we collected a set of target words to be used in the LSSD task. We compared several distributional semantics methods, and showed that using word embeddings in a modified SVM kernel achieves the best results (an increase of 10%F1 and 8%F1 for S vs.L and

S vs. *Lsent* disambiguation task, respectively, against a SVM baseline). The resulting datasets were made available to the research community.

10.3 The Role of Conversation Context for Sarcasm Detection

10.3.1 Introduction

Computational models for sarcasm detection have often relied on the content of utterances in isolation. However, speaker’s sarcastic intent is not always obvious without additional context. Focusing on social media discussions, in this line of work (Ghosh et al., 2017) we investigated two issues: (1) does modeling of conversation context help in sarcasm detection and (2) can we understand what part of conversation context triggered the sarcastic reply. To address the first issue, we investigated several types of Long Short-Term Memory (LSTM) networks that can model both the conversation context and the sarcastic response. We show that the conditional LSTM network (Rocktäschel et al., 2015) and LSTM networks with sentence level attention on context and response outperform the LSTM model that reads only the response. To address the second issue, we present a qualitative analysis of attention weights produced by the LSTM models with attention and discuss the results compared with human performance on the task.

10.3.2 Methods, Assumptions and Procedures

Datasets. One goal of our investigation is to comparatively study two types of social media platforms that have been considered individually for sarcasm detection: discussion forums and Twitter. **Discussion Forums:** Oraby et al. (2016) have introduced the Sarcasm Corpus V2, a subset of the Internet Argument Corpus that consists of discussion forum data. This corpus consists of sarcastic responses and their context (quotes to which the posts are replies to). The annotation of sarcastic vs. non-sarcastic replies was done using *crowdsourcing*, where annotators were asked to label a reply as sarcastic if any part of the reply contained sarcasm (thus annotation is done at the reply/comment level and not sentence level). The final gold sarcastic label was assigned only if a majority of the annotators labeled the reply as sarcastic. Although the dataset described by Oraby et al. (2016) consists of 9,400 posts, only 50% (4,692 altogether; balanced between sarcastic and non-sarcastic categories) of that corpus is currently available for research. **Twitter:** We collected our own corpus of conversational Tweets that contains 25,991 instances (12,215 instances for sarcastic class and 13,776 instances for the non-sarcastic class). We observe that 30% of the tweets have more than one tweet in the conversation context. There are two main differences between these two datasets that need to be acknowledged. First, discussion forum posts are much longer than Twitter messages. Second, the way the gold labels for the sarcastic class are obtained is different. In the discussion forum dataset the gold label is obtained via crowdsourcing, thus the gold label emphasizes whether the sarcastic intent is *perceived* by hearers (we do not know if the speaker intended to be sarcastic or not). In the Twitter dataset the gold label is given directly by the #hashtag the speaker used, signaling clearly the speaker’s sarcastic intent.

Models. To assess the effect of conversation context (c) on labeling a reply (r) as sarcastic or not sarcastic, we consider two binary classification tasks. We refer to sarcastic instances as S and non-sarcastic instances as $N S$. In the first task, classification is performed using the reply in isolation (S^r vs. $N S^r$ task). In the second, the classification considers both the reply and its context (S^{c+r} vs. $N S^{c+r}$ task). We experiment with two types of computational models: Support Vector Machines (SVM) with linguistically-motivated discrete features (such as bag of words, semantics and pragmatic feature, sarcasm indicator; used as baseline; SVM_{bl}), and approaches using distributed representations. For the latter we use the Long short-term Memory (LSTM) Networks (Hochreiter and Schmidhuber, 1997b) that have been shown to be successful in various NLP tasks. Since our goal is to explore the role of contextual information (our *first input*) for recognizing whether the reply (our *second input*) is sarcastic or not, we argue that using two LSTMs one that reads the context and one that reads the reply are a natural modeling choice. We propose LSTM networks

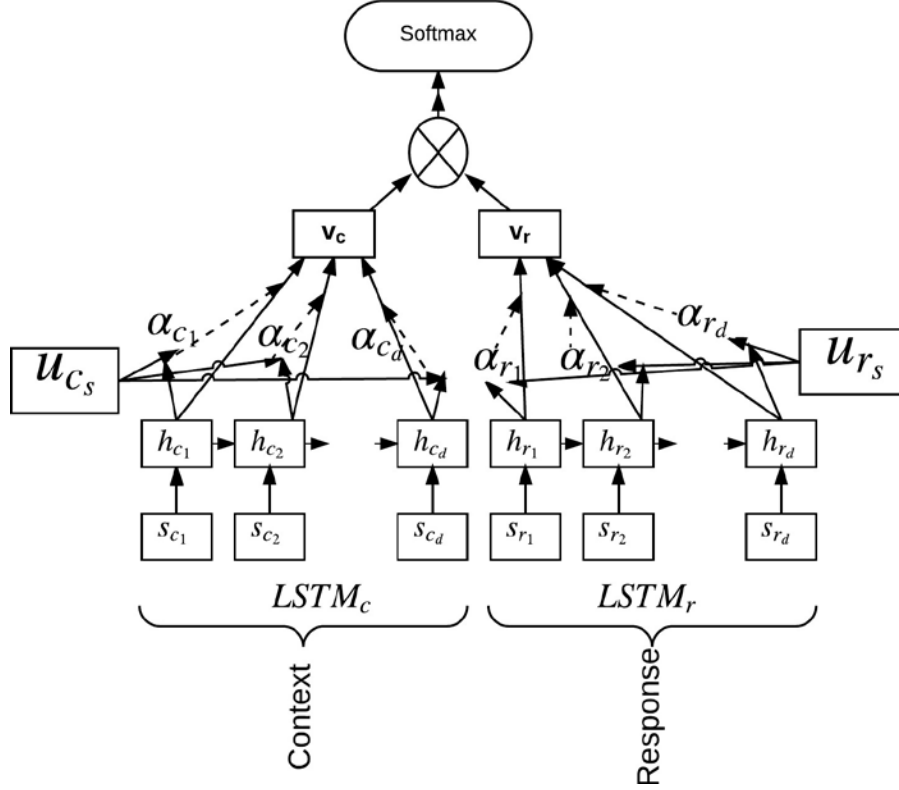


Figure 11: Sentence-level Attention Network for Context and Reply. Figure is inspired by (Yang et al., 2016)

with attention both at word and sentence level. Figure 11 shows the high-level structure of the attention model. The context (left) is read by an LSTM ($LSTM_c$) whereas the response (right) is read by another LSTM ($LSTM_r$). For the sentence level attention we represent each sentence by the average of its word embeddings. We also experiment with the *conditional encoding* model as introduced by (Rocktäschel et al., 2015) for the task of recognizing textual entailment. In this architecture, two separate LSTMs are used – $LSTM_c$ and $LSTM_r$ – similar to the previous architecture without any attention, but for $LSTM_r$, its memory state is initialized with the last cell state of $LSTM_c$. In other words, $LSTM_r$ is conditioned on the representation of $LSTM_c$ that is built on the context.

10.3.3 Results and Discussion

We report Precision (P), Recall (R), and F1 scores on S and NS classes. SVM_{bl}^r and SVM_{bl}^{c+r} respectively represent the performance of the SVM model using discrete features when using only the reply and the reply together with context. $LSTM^{ca}$ and $LSTM^{ra}$ are the attention-based LSTM models of context and reply, where the w , s and $w+s$ subscripts denote the word-level, sentence-level or word and sentence level attentions. $LSTM^{conditional}$ is the *conditional encoding* model (no attention). Table 37 shows the classification results on the discussion forum dataset. Table 38 shows the results on the Twitter dataset. Using contextual information helps, particularly when using $LSTM^{conditional}$ models or LSTM with sentence-level attention.

Qualitative Study. In this study we wanted to understand whether a specific *part of the context* triggers the sarcastic reply. To begin to address this issue, we conducted a qualitative study to understand whether (a) human annotators are able to identify parts of context that trigger the sarcastic reply and (b) attention

Table 37: Experimental results for the discussion forum dataset (**bold** are best scores)

Experiment	S			N S		
	P	R	F1	P	R	F1
SVM _{bl} ^r	65.55	66.67	66.10	66.10	64.96	65.52
SVM _{bl} ^{c+r}	63.32	61.97	62.63	62.77	64.10	63.5
LSTM ^r	67.90	66.23	67.1	67.08	68.80	67.93
LSTM ^c +LSTM ^r	66.19	79.49	72.23	74.33	59.40	66.03
LSTM ^{conditional}	70.03	76.92	73.32	74.41	67.10	70.56
LSTM ^{ras}	69.45	70.94	70.19	70.30	68.80	69.45
LSTM ^{cas} +LSTM ^{ras}	66.90	82.05	73.70	76.80	59.40	66.99
LSTM ^{caw+s} +LSTM ^{raw+s}	65.90	74.35	69.88	70.59	61.53	65.75

Table 38: Experimental results for Twitter dataset (**bold** are best scores)

Experiment	S			N S		
	P	R	F1	P	R	F1
SVM _{bl} ^r	64.20	64.95	64.57	69.0	68.30	68.7
SVM _{bl} ^{c+r}	65.64	65.86	65.75	70.11	69.91	70.0
LSTM ^r	73.25	58.72	65.19	61.47	75.44	67.74
LSTM ^c +LSTM ^r	70.89	67.95	69.39	64.94	68.03	66.45
LSTM ^{conditional}	76.08	76.53	76.30	72.93	72.44	72.68
LSTM ^{ras}	76.00	73.18	74.56	70.52	73.52	71.9
LSTM ^{cas} +LSTM ^{ras}	77.25	75.51	76.36	72.65	74.52	73.57
LSTM ^{caw} +LSTM ^{raw}	76.74	69.77	73.09	68.63	75.77	72.02
LSTM ^{caw+s} +LSTM ^{raw+s}	76.42	71.37	73.81	69.50	74.77	72.04

weights are able to signal similar information. For (a) we designed a crowdsourcing experiment and for (b) we looked at the attention weights of the LSTM networks. We first measure the overlap of Turkers choice with the attention weights. For the sentence-based attention model (i.e., LSTM^{cas}+LSTM^{ras} model for the discussion forum), we selected the sentence with highest attention weight and matched it to the sentence selected by Turkers using majority voting. We found that 41% of times the sentence with the highest attention weight is also the one picked by Turkers.

10.3.4 Conclusion

This research makes a complementary contribution to existing work of modeling context for sarcasm/irony detection by looking at a particular type of context, conversation context. We have addressed two issues: (1) does modeling of conversation context help in sarcasm detection and (2) can we determine what part of the conversation context triggered the sarcastic reply. To answer the first question, we show that Long Short-Term Memory (LSTM) networks that can model both the context and the sarcastic reply achieve better performance than LSTM networks that read only the reply. In particular, conditional LSTM networks and LSTM networks with sentence level attention achieved significant improvements (e.g., 6-1% F1 for discussion forums and Twitter messages). To address the second issue, we presented a qualitative analysis of attention weights produced by the LSTM models with attention, and discussed the results compared with human annotators. We also showed that attention-based models are able to identify inherent characteristics of sarcasm (i.e., sarcasm markers and sarcasm factors such as context incongruity).

Part III

Cognitive Relations

Chapter 11

Detecting Community Endorsement

11.1 Summary

This effort addressed the problem of detecting and ranking of community endorsed comments and participants in discussion threads. Detecting community endorsement provides a mechanism for characterizing entities in a knowledge base in terms of social status and a mechanism for characterizing beliefs in terms of their resonance with a particular community. The approach integrates language cues and discussion context, and explores approaches that can provide insights into model decisions.

11.2 Introduction

Community endorsement represents a form of influence or social recognition observed in social media, where in the course of a discussion a few people succinctly express views that reflect the beliefs or sympathies of a large number of people reading and/or participating in the discussion. A community endorsed person (CEP) can be thought of as an emergent spokesperson or thought leader. Community endorsement is rare – the vast majority of comments and discussion participants do not have strong endorsement. In addition, many discussions have no one that receives endorsement, either because the original post reflects a topic that few people care about or because the community is divided on an issue. In the knowledge base (KB), community endorsement is a property of an entity – a community endorsed person (CEP) is highly endorsed in a specific discussion – but it can also be a property of a belief (a specific comment is highly endorsed in a discussion). We use Reddit discussion forums as a platform for developing community endorsement algorithms, because we can avoid a costly annotation effort by using comment karma (roughly the sum of up and down votes by readers) as a proxy for community endorsement. Because the Reddit display emphasizes high-karma comments and posts, awarding karma is equivalent to encouraging others to read a particular thread. Another advantage of working with Reddit is that there are many different subreddits that reflect different online communities, through which we can gain an understanding of how language cues associated with high community endorsement vary with both topic and community.

Community endorsement is related to popularity, but many popularity studies simply look at the number of responses, shares or retweets (depending on the medium). While volume does play an important role, a comment can receive a large number of responses because it is controversial. By considering up and down votes, we have a more meaningful indicator of endorsement, but also a challenging task of integrating response structure, timing and complex language analysis. Community endorsement is reflected in the wording of the responses of participants to a person's comments (agreement, positive sentiment) and in the fact that their comments impact the direction of the discussion. However, the wording of the comment also plays a role – two people expressing the same point in different ways can get very different levels of endorsement.

The main goal of our work is detecting CEPs, but we also explored prediction of comment-level endorse-

ment for two reasons. First, we hypothesized (and later showed) that predictions at the level of individual comments would facilitate learning text-based cues to endorsement. Second, the most highly endorsed comments made by someone serves as evidence to back up KB assertions. The research addressed four main challenges: i) classifiers need to effectively integrate lexical cues with conversational context at both the comment and CEP level; ii) data-driven training is faced with the problem of highly skewed data, since highly endorsed people and comments are rare; iii) domain variability; and iv) a need for effective and interpretable representation of text to provide evidence for decisions.

11.3 Methods, Assumptions and Procedures

Data and Task Definition

We initially collected data from 16 subreddits on topics varying from science to politics to personal interests, with roughly 100k discussion threads in total and more than 2M comments. The comments in all discussion threads are awarded karma (positive or negative) by other commenters who appreciate the points made in a comment. A full collection of Reddit discussions from 2007 through May 2015 became available as a packaged download, which provides a much larger set of training data than we used in our earlier work. A limitation of this data was that it did not include the original post, which is critical for exploring language features associated with relevance of comments to a discussion. Thus, we downloaded the original posts for 8 subreddits reflecting fairly different communities, including ChangeMyView (used in experiments on social influence) and 7 others used in community endorsement work (askmen, askwomen, askscience, fitness, politics, worldnews, atheism). The data can be shared simply by providing URLs of the discussions.

To support research on automatic detection and ranking of people who receive strong community endorsement, we developed a mapping from the karma scores of all the participant's comments to a single CEP score and a criterion for determining the high CEP outliers, based on analysis of a collection of 6 subreddits. The data was divided (by time) into training and test sets, and annotated with CEP labels, and made available to CUNY and UTDallas for DEFT research. In addition, we implemented and shared scoring software for reporting detection and ranking performance, using F-score and normalized discounted cumulative gain (NDCG), respectively. We tried to propose an open evaluation that multiple teams were interested in, but LDC did not want to base the evaluation on Reddit data, and the cost of annotating other data was prohibitive.

Comment endorsement prediction

We propose multiple alternative tasks for exploring community endorsement at the comment level, with the goal of supporting the CEP detection work. Formulating karma prediction as a regression task is problematic, since the distribution is roughly Zipfian and we are mainly interested in the tail of the range. Our early work looked at ranking comments posted within a similar time window to control for the effect of time. This task was useful for analysis, but less so for CEP prediction. To address the challenge of a skewed distribution in a less constrained predictor, we developed a new proxy task of predicting quantized karma level using a karma binning strategy that is skewed and subreddit dependent. Most of our research built on this task, which assumed that the full discussion tree was available. Finally, we used one additional task that emphasized the current discussion context and comment, which involved use of reinforcement learning to identify the most highly endorsed subthreads of the discussion, again focusing on constrained time windows.

Community endorsed person detection

The fact that CEPs are rare (5% or less of participants) creates a challenge for machine learning. To ameliorate the problem, we developed an architecture that uses a series of two SVM classifiers that leverage different features, with the first tuned for high CEP recall using thread structure and timing features, and the second trained on the pruned lists and tuned to maximize F-score with the addition of text features. The

subset of participants who are labeled as a CEP are then ranked according to the level of endorsement using similar features. We explored three rankers (SVMrank, LambdaRank, and LambdaMART) and obtain the best performance with LambdaMART, which is a ranker based on a gradient boosting decision tree that has the advantage that it can directly optimize for NDCG. In initial work, text features of the first comment of an author are integrated directly with discussion context features in the stacked SVMs. In later work, we used a function accumulating the automatically estimated endorsement level from all of their comments in a discussion thread.

Approaches for characterizing text

We first explored knowledge-engineered features based on the literature associated with popularity prediction in social media, including reputation, informativeness, relevance, novelty, mood, community style matching, etc. We also introduce the idea of representing the K-index of an individual as a proxy for different types of influence, analogous to H-index for an author based on the paper citations. Subsequently, we developed a series of neural models for learning text embeddings. Three methods were developed based on the task of predicting quantized karma, including: an unsupervised learning strategy using a factored neural network model based on two LSTMs with attention mechanisms that separately represent global context, comment content, and comment intent (Cheng et al., 2017); and two supervised models (a gating model that learns a context-dependent weighting of text features to focus modeling effort on cases where text could make a difference (Fang et al., 2016), and a tree-structured LSTM to predict community endorsement of comments using forward and backwards propagation of information in the discussion tree (Zayats and Ostendorf, 2018). In addition, we explored bag-of-words embeddings in the context of the reinforcement learning task, where we try to learn relevance to popular topics in world news as a means of incorporating “world context” (He et al., 2017).

Domain differences

We explored domain (community) differences in three ways: i) analysis of feature importance in different domains using engineered features based on prior published research (Jaech et al., 2015), ii) reporting results for multiple domains (all studies), and iii) running cross-domain training experiments.

11.4 Results and Discussion

Endorsement (karma) analysis

Karma is associated with both comments and people, since people accumulate karma based on the karma of their comments. The distribution over both comments and people is highly skewed: most people and most comments are low karma. The distribution of karma for comments is roughly Zipfian within a subreddit, and subReddits with higher readership tend to have more karma, which can be accounted for with a simple scale factor. Thus, detecting that a comment is influential or has strong community endorsement is challenging because it is rare. We also find that even contributors with a high K-index make mostly low karma comments, so identifying the specific comments that are high karma is important for understanding why a particular person is being endorsed.

Feature exploration

As in related work on influence and popularity in discussion threads and on power in email, we find that graph structure is an important indicator of community endorsement. However, knowing the whole tree structure is not enough to pick the top karma comment. In experiments on lists of 30 comments, prediction accuracy using only tree structure and timing features is less than 40%, which suggests that analysis of language features should be useful. We conducted two series of experiments exploring knowledge-driven features. First, as a means of reducing the importance of timing and tree structure, we ran experiments identifying the highest karma comment submitted within a fixed window using a variety of timing, graph and text features motivated by prior studies of online communities. We obtained an accuracy of 24-31%

compared to a random baseline of 15-19%. In four of the six subreddits, the use of automatically selected text features gives a gain over relying on thread structure alone, but the particular types of features that are most informative (e.g. reputation vs. informativeness vs. mood etc.) vary substantially depending on the community. Pairwise high/low karma classification experiments point to a few indicators of low karma (short comments, use of pronouns such as “I”, “me”, “you”, low topical similarity to the post and parent comments). Pairwise positive/negative karma classification experiments expose community differences, e.g. swear words are positively received in some communities but negatively received in others. Verbs that reflect emotional intensity are often found in high karma comments. Further details are provided in (Jaech et al., 2015).

In the above experiments, the community style matching feature was not as useful as we anticipated based on related literature. To better understand this, we ran experiments on subreddit classification as a proxy for style matching, and then looked at how the models scored aligned with endorsement. We explored topic vs. style models, where the style models build on results from genre classification. The best topic model used a k-nearest neighbor (KNN) model combined with latent Dirichlet analysis, where the KNN model is helpful in subreddits which cover a range of topics. The most effective model for community style used a simple n-gram based on a mixed vocabulary of part-of-speech (POS) tags and a small number of discriminative words. The best results were obtained with the style model, which identifies the subreddit of a discussion with accuracy of 86.5%, compared to roughly 70% obtained by both the topic model and the n-gram language model used in our previous study. As hoped, the community style match score of a person has a statistically significant correlation with the endorsement status of that person (0.3-0.45) and the style match of the participants in a discussion as a group is correlated to the endorsement of the post, particularly for the politics subreddit (which also benefited most from the style feature in (Jaech et al., 2015)). While the topic information is not useful for predicting CEP status, it could potentially be used to generate a phrase summarizing the discussion to enhance KB assertions. Further details are described in (Tran and Ostendorf, 2016). Some of our subsequent work builds on the idea of characterizing community style with neural embeddings.

Most of our neural feature exploration experiments were in the context of comment endorsement, described in the next section. However, we also explored a method for learning comment embeddings via a reinforcement learning algorithm for the task of picking the highest karma comments in a fixed time window. In this case, the objective was to learn comment embeddings that take into account “world context,” specifically recent discussions on current events in another forum. In this model (He et al., 2017), external discussion threads are captured using a mechanism somewhat like a memory network, but the attention model for determining what is relevant considers timing and popularity as well as semantic similarity. Reinforcement learning is used to determine how to combine the current discussion history with world context in order to identify highly endorsed comments. In experiments with five different communities and using the worldnews subreddit to provide context, we find that the system learns different attention weights for the various factors depending on the community. For example, discussions in the news in the past day are most relevant for askscience, whereas the past week are more useful for other subreddits. Popularity is useful for askscience and politics, but not for todayilearned.

Comment endorsement results

We conducted a series of experiments predicting endorsement of comments as characterized by quantized karma. Three different modeling frameworks were explored emphasizing different aspects of the role of context in characterizing text. The different studies are not directly comparable due to some experimentation changes that were made between studies and the very different approaches, but together they provide insights into the problem that provide a good foundation for future work. Because of the greater computational requirements for training these models, experiments used only three subreddits: askmen, askwomen and politics.

The first study used a neural network classifier to combine submission context (discussion tree and timing features) with an LSTM-based comment embedding learned in an end-to-end fashion. Submission context was modeled with automatically learned latent basis vectors and used to provide a gating function that effectively put more or less weight on the text features depending on the context. The approach was motivated by observations leading to the hypothesis that early in the discussion, timing is more important than the language of the comment, whereas text matters more for comments late in the discussion. An advantage of using the latent basis vector representation is that they can be projected back to the original features, making it possible to learn interpretable context features that identify different types of high and low endorsement comments. In experiments on three subreddits, we find that context-dependent gating indeed makes text features more useful for prediction (Fang et al., 2016).

In the above model, which uses end-to-end learning, the text embedding is necessarily relatively simple. In a second study, we explored a model that explicitly represents discourse context and uses unsupervised learning, which can take advantage of more data. We introduce a factored neural network model based on LSTMs with attention mechanisms that separately represent global context, comment content, and response generation characteristics of a comment (Cheng et al., 2017). These embeddings were shown to be an improvement over simple LSTM comment embeddings as well as other unsupervised methods for learning comment embeddings, including latent Dirichlet analysis (Blei et al., 2003a) and the doc2vec model (Le and Mikolov, 2014). As in the previous work, we use latent basis vectors, but in this case to model global context. The basis vectors effectively learn community style and, unlike other unsupervised approaches, combining this with the attention mechanism allows the model to learn position-dependent style. To better understand the impact of the different components of the model, we conducted analyses of the embeddings, finding that the context model learns structured characteristics of writing style of the community, particularly comment openings, and that the response generation component captures aspects of a comment that seem to trigger positive vs. negative sentiment in its replies. The unsupervised embeddings are used in four different classifiers, assessing multi-class logistic regression vs. ordinal regression and shallow vs. deep models. The best results across three subreddits are obtained with deep ordinal regression, where adding text embeddings give a gain of 4-8% in performance over a baseline using just submission context features (Cheng et al., 2017).

In the study described above, the global context component represents the subreddit community, and the response component uses responses only in training in a form of multi-task learning that drives the comment embeddings to characterize factors relevant to responses. The textual context of a comment (e.g. the comment it is responding to, or the responses to it) is not explicitly represented. Intuitively, the textual context should be important, but it has been challenging to learn the salient features for endorsement prediction. The final model that we explore explicitly represents a comment in the context of the discussion history and response characteristics via a graph-structured LSTM that includes both the tree structure of comments and responses and the temporal structure of the sequence of comments replying to the same parent (Zayats and Ostendorf, 2018). Information is propagated using forward and backward (or upward) passes in the discussion tree. The approach seeks to model our observation that popular sub-threads in a discussion have comments that may be benefiting from each other in terms of karma. Results show that incorporating context in this way provides a more effective use of text, giving a gain of 6-8% in performance over treating each comment independently. Analyses show that the backwards pass is particularly helpful in identifying controversial comments (correcting over-prediction of karma) and comments that are well received late in the discussion (correcting under-prediction of karma) (Zayats and Ostendorf, 2018).

CEP detection results

In the first year of the effort, we conducted a dry-run evaluation with CUNY and UTDallas evaluating on 7 different online communities (subreddits). We achieved the best results for 6 of the 7 domains for the binary problem of detecting CEPs, with F-scores ranging from 35-54% across the different domains (compared to

a baseline of 22-28%). We were the only group reporting on the ranking task; our NDCG scores ranged from 53-88% across the 7 communities. Since there was not much interest in the ranking task by other sites, our subsequent efforts focused on detection evaluated with F-scores.

Our early work used SVMs with submission context and a subset of the engineered comment features (based on forward feature selection) associated with the first post of the discussant. This approach fails to take advantage of the information available when a discussant makes multiple posts, and the text features were providing little benefit over the submission context features. For this reason, we started focusing on comment endorsement prediction. Once we had obtained good results with some of our neural comment embeddings, we began exploring mechanisms to integrate comment endorsement predictions into CEP detection. Using the quantized karma predictions made it possible to focus on the highest karma comments for extracting graph and timing features, and to obtain average statistics over multiple comments by an author. Using the comment endorsement predictions from the graph-LSTM described above, we obtain a gain of roughly 10% in F-score over the system using submission context alone, with F-scores of 44-46% on three subreddits (askmen, askwomen and politics). Adding the Q-value associated with reinforcement learning as another feature did not give significant additional performance gain. We expect that the comment predictions will also be useful for improving confidence scores and providing evidence for CEP predictions.

Domain differences

As mentioned above, domain (subreddit) differences were observed in the types of features that were most relevant for a community. In experiments with different communities, we observed substantial differences in performance. In the pilot evaluation, we obtained the best result for askscience (F=54.4), the worst results for fitness (F=35.1) and five other subreddits (askmen, askwomen, atheism, politics, worldnews) all getting F-scores in the 39-42 range. Subsequent experiments with an improved classifier had the same outliers, with greater variation among the other subreddits. Some of the variability is due to style differences in the communities, but differences in the size of the communities also seem to play a role.

Experiments aimed at comment endorsement with text embeddings focused on three domains that were in the middle range but reflected very different community styles: askmen, askwomen, and politics. Differences in performance held across the different methods explored. Politics was the most difficult to predict, with macro-average F-score of 50 for our best methods, compared to 55-56 for askmen and askwomen.

The considerable variability across domains raises the question of how effective algorithms trained on one or more domains will be on a new domain. To understand the impact, we ran cross-domain testing with the original. We explored mismatched train/test conditions with seven different subreddits, finding that: i) two domains (askmen and atheism) are outliers in that performance severely degrades with mismatched training (30-45% reduction), and ii) two domains (askwomen and worldnews) seem to be quite useful for all other domains that we tested on (at most a 6% degradation for worldnews). These results point to the potential for rapid generalization by leveraging a few key sources to represent style and topics of current interest.

11.4.1 Resulting Resources

Algorithm delivery: We delivered a CEP detection algorithm to ADEPT and passed all tests. The algorithm includes a confidence estimate associated with CEP prediction, which uses an automatically learned warping of the CEP posteriors trained on development data. The algorithm that was delivered represents an early stage of the research. Later developments are available open source on github, as noted in publications.

Knowledge base population: Software was implemented to make assertions about entities in the knowledge base (KB), specifying that someone is strongly endorsed in a particular discussion with a link to that discussion. Assertions have been made in English and Spanish based on an intermediate version of the system.

Foreign languages: We implemented a baseline CEP system in Spanish. It is separately trained, but it

uses all the same software infrastructure as the English system with the exception of a small amount of text pre-processing. Performance on Spanish is lower than on English, primarily because these subreddits have much smaller readership, so discussions are shorter and there are fewer highly endorsed participants.

11.5 Conclusions

In summary, this effort introduced the problem of detecting and ranking of community endorsed comments and participants in discussion threads. Detecting community endorsement provides a mechanism for characterizing entities in a knowledge base in terms of social status and a mechanism for characterizing beliefs in terms of their resonance with a particular community.

We explored a variety of approaches on multiple domains (online communities) integrating submission context and textual cues. For the domains that we focused on (askmen, askwomen, politics), we found that comment endorsement level could be predicted with macro F-score in the range of 50-56, and CEP prediction with our best model obtains F-scores of 44-46. The politics discussions are the most difficult, but experiments leveraging the context of worldnews provided the greatest gain for the politics subreddit, which is promising for future work. The cross-domain experiments also demonstrate the potential for learning from selected domains to provide good performance on a new domain.

Chapter 12

Influence

12.1 Summary

We have developed methods for identifying situational and persuasive influence in social media.

Situational influence refers to influence created over the course of conversation through an individual's posts. We performed experiments on training predictive models of discourse characteristics (e.g., agreement and dialog patterns), dialectics (e.g., claims and argumentation), and characteristics of participants not dependent on the conversation (e.g., author traits and credibility). We then used features derived from these characteristics to improve the identification of influencers.

We also consider persuasive influence. Whereas situational influence is a relationship between one person and other members of a group, persuasion is a binary relationship that requires one person to change the mind of another person. We build neural models for persuasiveness prediction and demonstrate improvements over a strong baseline that uses Jaccard similarity between content and stop words in the two users' posts.

12.2 Situational Influence

12.2.1 Introduction

Automatic identification of situational influence will enable recognizing how influential a person is within a specific situation and how the person's influence varies across discussions. The abundance of content available on the web allows us to analyze the way people interact and the roles they play in a conversation on a large scale. One such role is that of influencer; in many conversations, there is one person who influences others. An influencer can alter the opinions of their audience, resolve disagreements where no one else can, be recognized by others as one who makes important contributions, and often continue to influence a group even when not present. Other conversational participants often adopt their ideas and even the words they use to express their ideas.

An example of an influencer in a Wikipedia Talk Page discussion regarding an Image for Death is shown in Table 12 where Richard001 was annotated as the influencer. Richard001 initiates the thread and maintains participation throughout. He gets support for his ideas about changing pictures and debates with dissenters. Opposition to his opinions are mild as well.

In our experiments, influence is predicted per person within a discussion. In other words, for each participant X in each thread Y, the system answers the following question: Is X an influencer in Y? We detect influence across five online data sources that have been annotated for influence: Wikipedia Talk Pages, LiveJournal weblogs, Political Forum discussions, Create Debate debate discussions, and Twitter microblog conversations.

P1 by Richard001 Two recently added images include a dead dog, which I find random and in somewhat bad taste, and an image of the pope, along with caption making POV assertions about his entrance to the ‘afterlife’. Can we please have discussion before changing the image.
P2 by Bleh999 The Dog is clearly the best illustration of death put forth so far imho, the pope one is indeed POV. Why do you think the dog is bad taste? P3 by Richard001 I’ve added small thumbs of each of the images we have had recently. As for the dead dog, it’s not particularly appealing to me. There are plenty of other images we could use as well, the important thing is to discuss the changes.
P4 by Ksyrie I am for your remarks–
P5 by Bleh999 Note the image was only used thumbnail size in the article so if you were so offended why did you click so to see the larger version?
P6 by Richard001 I don’t think it’s a valid point that it’s small - would it be okay to have a thumbnail of a detailed photograph of two people enthusiastically copulating on the sexual intercourse article if you couldn’t see their genitalia in the thumbnail? I strongly suggest using an image from long enough ago that it won’t cause any edit disputes - the Mongol invasions for example.
P8 by Ksyrie I doubt if any available pic far away too 1000 years ago, a slaughter scene from Sparta War?–
P9 by Bleh999 Once again you are wrong, take a look at wikipedia commons there are graphic images of sexual acts available
P10 by Fusion7 I am rather upset with the current image Image:Civil War graves.JPG This article should be mainly about the science of death, and its ties with human culture secondary. However, this image does the opposite. It focuses not on death itself, but our traditions that revolve around its occurrence.
P11 by Fusion7 I agree with your comments

Figure 12: Influence Example: A Portion of a Wikipedia Talk Page Discussion Regarding an Image for Death Displaying Richard001 as the Influencer. Replies Indicated by Indentation (e.g., P2 is a Response to P1)

12.2.2 Methods, Assumptions and Procedures

Our approach is a supervised method using system components based on Opinion, Claims, Argumentation, Persuasion, Agreement, Author Traits, Dialog Patterns, and Credibility to detect situational influence across online genres (Rosenthal and McKeown, 2017). Each of these learned components computes characteristics of the conversation based on analysis of language within the thread and the participation of the potential influencer. We demonstrate how a sophisticated learned analysis of discourse characteristics (e.g., agreement and dialog patterns), dialectics (e.g., claims and argumentation), and characteristics of participants not dependent on the conversation (e.g., author traits and credibility) can be exploited to detect influencers.

Persuasion, Claim, and Argumentation. The first set of features relate to persuasion. In this work, we focus on attempts to persuade (meaning we disregard the effects of whether they were actually persuasive) someone to change their opinion. An attempt to **persuade** consists of two parts: making a claim and closely following it with an argumentation. A sentence is considered to be a **claim** if it is opinionated and the author expresses a belief. An **argumentation** is a justification to a claim.

We automatically label the sentences that are claims (Rosenthal and McKeown, 2012) and argumentations in a discussion. The claim system is a supervised system that uses features such as opinion, committed belief, Parts-of-Speech (POS), and n-grams to detect whether a sentence is a claim. The argumentation

system exploits Rhetorical Structure Theory (RST) relations in the RST Penn Treebank to extract lists of indicators of relations and co-occurring content word pairs for each of the indicators (using Wikipedia). We define persuasion as a claim followed by an argumentation. In our system, we take the first argumentation following the claim as the attempt to persuade.

The **claim** system is trained and evaluated on two datasets: 285 LiveJournal blogposts and 51 Wikipedia discussion forums. Each dataset consists of 2,000 sentences that are between 30-120 characters. Each of our datasets were also annotated for subjectivity. The statistics of each corpus are described in figure 13.

Corpus	Claims	Not Claims	Subjective Phrases	Objective Phrases	Vocabulary Size
LiveJournal	1197 (60%)	791 (40%)	3035 (39%)	4709 (61%)	4747
Wikipedia 1282	(64%)	715 (36%)	1319 (37%)	4496 (63%)	4342

Figure 13: Statistics for each corpus; LiveJournal and Wikipedia

We train a supervised machine learning model to identify claims using lexical features along with features based on **sentiment**, committed **belief**, and **social media**. We train a **sentiment** system to detect subjective vs. objective phrases, which includes social media features such as emoticons, acronyms, and misspellings to adapt it towards our corpora. We train the sentiment system on the same sentences as our claim corpus and use the output of the sentiment system to compute three features. First, we use it to determine whether sentiment exists in the sentence, second to determine the ratio of the sentence that is subjective, and lastly, the count for each of the subjective/objective patterns of the sentence using 1-3 chunks. We also use the system described in Prabhakaran et al (2010) to extract words in the sentence that express **belief**: committed (“I know”), non-committed (“I may”), and not applicable (“I wish”) which vary in intensity from strong to weak respectively. Furthermore, in order to take into account the nature of online corpora, we compute “**social media** features” from acronyms, emoticons, slang, capitalization, and punctuation. Finally, we include Part-of-Speech tags and n-gram features (1-3 words) and performed feature selection using a χ^2 test. Sentences that are questions are often not claims so we use whether the sentence ends in a question mark (?) as a binary feature. We use the majority class and the question feature as our baselines. We present results of 10-fold cross validation and using one dataset as training and another as testing. The full set of results is shown in Figure 14.

The output of the claim and argumentation systems is then used to generate the features for claim, argumentation, and attempts to persuade. We include features to indicate persuasiveness (indicating a feature generated for both claim, argumentation, and persuasion) by examining whether the participant is persuasive, total occurrence of persuasion, all/number of persuasive posts, and the post with the most attempts to persuade. The total occurrence of each feature per person is normalized by post size and/or sentence size in the thread where applicable (e.g., claims occur on the sentence level and are therefore normalized by sentence size). In addition, we also include a feature to indicate the number of times the first sentence in the post was an argumentation or a claim normalized by the number of posts by the participant. Starting a post with an argument or claim can indicate that it is stronger. Finally, we also include Bag-of-Words from the text of the argumentative and claim sentences. This represents meaningful content that the participant has written.

Agreement. A person who agrees with others is indicating they are following someone else’s opinions. This can imply that they are less likely to be influential. On the other hand, disagreeing can indicate influence because the person has their own conflicting opinions they are trying to push. A lack of (dis)agreement can indicate that the post is not relevant and not influential.

The agreement component (Rosenthal and McKeown, 2015) is a supervised classifier using **stylistic** and **meta-thread** features as well as more complex features such as **accommodation**, **sentiment** (specifically

Experiment	Cross-validation		Cross-domain	
	LJ	Wiki	L-W	W-L
Majority	60.2	64.2	64.2	60.2
?	64.1	69.3	64.1	69.3
?+sm	64.5	69.7	64.7	69.9
?+sentiment	65.5	69.4	65.5	69.3
?+belief	63	69.6	66.1	73.7
?+n-grams	65.6	73.3	70.5	77.4
?+pos	67.6	70.6	70.4	72.4
?+sentiment + belief	65.9	69.5	69.0	73.5
?+belief+pos	67.6	72.1	72.4	77.4
?+sm+pos+n-grams	67.4	74.7	73.7	79.4
?+sentiment+n-grams+pos	67.4	73.9	70.7	77.6
?+sentiment+belief+n-grams+pos	66.9	73.1	72.1	80.0
?+ belief+sm+pos+n-grams	67.3	73.4	75.6	82.2
All	66.9	74.7	72.8	80.8

Figure 14: Accuracy for cross-validation within each corpus and cross-domain between each corpus. We experimented training on LiveJournal and testing on Wikipedia (L-W) and training on Wikipedia and testing on LiveJournal (W-L). We use two baselines: The majority class, and the question feature. The features used are question (?), social media (sm), sentiment, belief, n-grams, and pos.

Dataset	Thread Count	Post Count	Agree	Disagree	None
ABCD	9981	185479	38195	60991	86293
IAC	1220	5940	428	1236	4276
AWTP	50	822	38	148	636

Figure 15: Statistics for full datasets

polarity), and sentence **similarity** to predict whether each post in the discussion agrees or disagrees with the post it is responding to. It is also possible for a person to neither agree nor disagree with those posts.

We use three corpora to train and evaluate our agreement model: the Internet Argument Corpus (IAC), Agreement in Wikipedia Talk Pages (AWTP), and Agreement by Create Debaters (ABCD), a new corpus created using self-labeling. The IAC consists of posts gathered from 4forums.com discussions that were annotated on Mechanical Turk and is labeled using a scale of [-5, 5] where -5 indicated high disagreement, 0 no (dis)agreement, and 5 high agreement. We converted the scalar values to (dis)agreement with [-5, -2] as disagreement, [-1, 1] as none, and [2, 5] as agreement. The AWTP corpus is Wikipedia talk pages (used to discuss edits) labeled at the sentence level for agreement, disagreement, or neither. In the Create Debate corpus, the participants in the debate choose what side they are on each time they participate in the discussion. We create the ABCD corpus as follows: the side label determines whether a post (the Response) is in agreement with the post prior to it (the Quote). If the two labels are the same, then they agree. If the two labels are different, they disagree. When the author is the same for both posts, there is no (dis)agreement as the second post is just a continuation of the first. Finally, the first post and its direct responses do not agree with anyone; the first post does not have a side as it is generally a question asking whether people are for, or against the topic of the debate. Statistics on these datasets are presented in Figure 15

We train a logistic regression classifier to predict one of three labels: agree, disagree, or none between a post and its response. We use features based on thread structure, style, sentiment, similarity, and accomo-

dation. **Meta-thread** structure features include whether the post is the root of the discussion, the reply was by the same author, the distance, or depth, of the post from the beginning of the discussion, and the number of sentences in the response. **Stylistic** features include features derived from the Linguistic Inquiry Word Count (LIWC) and features based on punctuation, capitalization, acronyms, emoticons, and word lengthening. **Sentiment** features such as the presence of polarity or positive/negative sentiment are derived from the response using a sentiment classifier to identify opinionated sentences. Latent sentence **similarity** features are used to measure whether a post and response are discussing the same topic and are also combined with sentiment features. Finally, **accommodation** features are created to determine whether two posts use the same manner of speaking based on lexical and syntactic similarity. These features are combined with n-grams and question features (whether the sentence ends with a question mark).

We present the results of experiments using the IAC, AWTP, and ABCD corpora. In Figure 16, we present results training on 80% of the ABCD and testing on 10% (while using another 10%) for development. In Figure 17, we train on the ABCD and IAC and test on the IAC. In Figure 18, we train on the ABC and IAC and test on the AWTP (because the AWTP is very small). We compare our experiments to two baselines. The first is the majority class, which is none. The second baseline is n-grams, the commonly used baseline in prior work. We compute statistical significance using the Approximate Randomization test, a suitable significance metric for F-score.

Features	None	Agree	Disagree	Avg
majority	63.2	0.0	0.0	21.1
n-gram	45.7	35.6	41.3	40.9
n-grams+POS+lex.-style+LIWC in R	58.7	42.2	51.6	50.8
Thread Structure	100	45.8	62.0	69.2
Accommodation	74.0	45.1	59.1	59.4
Thread+Accommodation	99.6	57.8	68.2	75.2
All	99.6	58.0	73.1	76.9
Best	100	58.5	73.0	77.6

Figure 16: The effect, in F-score, of conversational structure in the ABCD corpus. Statistical significance is shown over majority (α) and n-gram (β) baselines.

Features	IAC				ABCD			
	None	Agree	Disagree	Average	None	Agree	Disagree	Average
majority	85.1	0.0	0.0	28.4	85.1	0.0	0.0	28.4
n-gram	58.6	11.7	27.8	32.7	46.7	7.8	36.6	30.3
n-grams+POS+lexical-style+LIWC in R	54.1	12.0 ^{α}	29.7 ^{α}	31.9	43.9	13.6 ^{α}	30.1 ^{α}	29.2
Thread Structure	87.4 ^{β}	25.3 ^{$\alpha\beta$}	50.0 ^{$\alpha\beta$}	54.2 ^{β}	87.3 ^{β}	26.4 ^{$\alpha\beta$}	53.8 ^{$\alpha\beta$}	55.8 ^{β}
Accommodation	52.9	13.9 ^{α}	32.4 ^{α}	33.1	51.7	14.7 ^{α}	34.3 ^{α}	33.6
Thread+Accommodation	87.5 ^{β}	26.5 ^{$\alpha\beta$}	48.9 ^{β}	54.3 ^{$\alpha\beta$}	87.2 ^{β}	28.0 ^{$\alpha\beta$}	55.5 ^{$\alpha\beta$}	56.9 ^{β}
All	83.5 ^{β}	28.8 ^{$\alpha\beta$}	50.4 ^{$\alpha\beta$}	54.2 ^{β}	87.3 ^{β}	27.0 ^{$\alpha\beta$}	41.2 ^{α}	51.8
Best	87.4 ^{β}	31.5 ^{$\alpha\beta$}	54.4 ^{$\alpha\beta$}	57.8 ^{β}	87.3 ^{β}	25.5 ^{$\alpha\beta$}	57.3 ^{$\alpha\beta$}	56.7 ^{β}

Figure 17: The effect, in F-score, of conversational structure in the IAC test set using the IAC and ABCD as training data. Results highlighted to indicate statistical significance over majority (α) and n-gram (β) baselines.

The output of the agreement system is used to generate features to indicate how often each person is

Features	IAC				ABCD			
	None	Agree	Disagree	Average	None	Agree	Disagree	Average
majority	87.2	0.0	0.0	29.1	87.2	0.0	0.0	29.1
n-gram	68.1	12.7	21.3	34.1	36.5	11.6	32	26.7
n-grams+POS+lexical-style+LIWC in R	64.1	12.1 ^{α}	22.7 ^{α}	33.0	54.0 ^{β}	27.7 ^{$\alpha\beta$}	36.2 ^{$\alpha\beta$}	39.3 ^{β}
Thread Structure	58.0	12.4 ^{α}	23.7 ^{α}	31.4	63.6 ^{β}	15.0 ^{α}	33.4 ^{α}	37.3
Accommodation	52.4	12.4 ^{α}	30.7 ^{$\alpha\beta$}	31.8	50.7 ^{β}	17.5 ^{$\alpha\beta$}	40.1 ^{$\alpha\beta$}	36.1 ^{β}
Thread+Accommodation	55.0	14.9 ^{α}	37.2 ^{$\alpha\beta$}	35.7	62.9 ^{β}	21.3 ^{$\alpha\beta$}	52.2 ^{$\alpha\beta$}	43.9 ^{β}
All	64.2	15.5 ^{α}	36.4 ^{$\alpha\beta$}	38.7	61.9 ^{β}	25.8 ^{$\alpha\beta$}	43.5 ^{$\alpha\beta$}	43.7 ^{β}
Best	59.3	14.4 ^{α}	34.5 ^{$\alpha\beta$}	36.1	63.6 ^{β}	23.3 ^{$\alpha\beta$}	46.8 ^{$\alpha\beta$}	44.4 ^{β}

Figure 18: The effect, in F-score, of conversational structure in the AWTP test set using the IAC and ABCD as training data. Statistical significance is shown over majority (α) and n-gram (β) baselines.

(dis)agreed with (from) and how often they (dis)agree with others (to). We also have features to indicate what the first to and from post are as a person’s initial position (from) and first impression (to) in the discussion can have a greater impact than later posts.

Author Traits. The psychological phenomenon of social proof entails that a person will be influenced by others in their surroundings. The social proof phenomenon is apparent when a person attempts to reflect the behavior of others. This occurs because people tend to assume, whether true or false, that others know more about the situation at hand. One manner in which people can be similar is through shared author traits such as demographics as well as political affiliation. Thus, social proof strongly motivates the inclusion of author traits in influence detection.

The author trait component is a single supervised classifier that uses **stylistic** and online **behavioral** features (Rosenthal and McKeown, 2016). The classifier is trained on different data types (e.g., age and gender) to generate individual models used to predict each author trait. Our author trait data comes from two different types of online sources; weblogs for age and gender and microblogs for politics and religion. We use the publicly available blogger.com authorship corpus and the LiveJournal age corpus to detect age and gender. There are several websites that either automatically generate (tweepz.com), or allow users to self-label (twellow.com and wefollow.com) their Twitter account into categories. We download Twitter users based on political party (Republican/Democrat), and religion (Christian, Jewish, Muslim, Atheist). We present statistics on these datasets in Figure 19.

We implemented an author trait detection system that uses behavioral and stylistic features to automatically detect author traits such as demographics and political affiliations. We build models for each author trait individually, applying a Linear Regression to predict age and a logistic regression to predict all other traits. For our **stylistic** features, we use Linguistic Inquiry Word Count categories and features derived from statistics on capitalization, punctuation, word lengthening, and acronyms. Online **behavioral** features include the most common hour and most common day of the week that the person posts. These features are combined with n-gram and part-of-speech features. Results are reported in Figure 20.

The output of the models is used to generate three groups of features for detecting influence related to each of the four author traits: age, gender, religion, and political party. The groups of features are **single** features, **majority** features, and **combination** features. The **single** features are marked as true for the predicted trait (age is the exact year as well as older or younger than 1982). The **majority** features are determined by whether a person is predicted to be in the majority within the discussion and have a majority feature corresponding to each author trait as well as whether the participant is in the majority in all author traits or in no author traits. We also explore whether the **combination** of author traits is beneficial. There have

author trait	source	label	size
age	blogger.com	year of birth	19098
	livejournal.com	year of birth	21467
gender	blogger.com	Male	9552
		Female	9546
	livejournal.com	Male	4249
		Female	3287
political party	Twitter.com	Republican	1247
		Democrat	1200
religion	Twitter.com	Christian	5207
		Islam	1901
		Atheist	1815
		Judaism	1486

Figure 19: The size (in users) of each trait corpus

Author Trait	Majority	Accuracy
Age	57.1	79.6
Gender	51.9	76.4
Political Party	51.3	75.2
Religion	50.0	78.3

Figure 20: The author trait results for accuracy

been many studies that show that certain tendencies towards an issue are based on many author traits. For example, women tend to vote for Democrats. We thus combine two author traits into a binary feature and majority feature.

Credibility. People are more likely to be influenced by those who have established credibility. Although considered a component, credibility is not a stand-alone system. Rather, we provide several features for predicting the influencers related to determining the credibility or lack thereof of each person in a conversation. Briefly, the features are as follows:

- **Grounding:** providing sources to back up claims through quotes, links, and numbers.
- **Name Mentions:** how often a participant refers to other participants (mentions) and how often others refer to the participant (mentioned)
- **Honorifics:** how often a person uses honorifics in their posts (e.g., President)
- **Out of Vocabulary:** the amount of unrecognized (and often informal) speech by the participant
- **Inquisitiveness:** the percentage of the participant's sentences that are questions, did the participant ask a question, and the person who had the most questions.

Dialog Patterns. The dialog component is a group of features used directly within influence detection and generated from the dialog structure of the thread. The features are as follows:

- **Initiative:** The participant is or is not the first poster of the thread
- **Last:** The participant is the last poster in the thread

- **Irrelevance:** The percentage of the participant's posts that were not replied to
- **Incitation:** The length of the longest branch of posts following the participant's posts
- **Investment:** The participant's percentage of all posts in the thread
- **Interjection:** The point in the thread at which the participant enters the discussion
- **Max Posts:** True if the participant has the most posts
- **Longest Post:** True if the participant has the longest post
- **Consecutive Post:** True if the participant has consecutive posts
- **Repetition:** True if the participant has two or more posts
- **Response Time:** The average period of time, in minutes, to receive a response
- **Active Time:** The amount of time, in minutes, the participant was active in the conversation (last post time - first post time)
- **Once:** The participant is part of the conversation only once

Each of these dialog patterns is motivated mainly by social proof. For example, people will often follow the person who has the initiative, investing often can increase the likelihood of being followed, and irrelevance can indicate a lack of social proof.

12.2.3 Results and Discussion

Online discussions can vary significantly in style, content, and audience. Our datasets target these variations to obtain a more comprehensive analysis of online social media. Our datasets are as follows:

- **LiveJournal** is a virtual community in which people write about their personal experiences in a weblog and can typically be considered as a public diary.
- **Wikipedia Talk Pages** are the talk pages associated with each Wikipedia Page where proposed edits for the page are discussed.
- **Create Debate** is a discussion forum whose format is a debate style where people can post on either a for or against side of an argument.
- **Political Forum** is a discussion forum website used to discuss political topics.
- **Twitter** is the most popular micro-blogging site on the web where people write short blurbs between 0 and 140 characters. A threading structure can be generated using tweets via retweets (tweeting something written by someone else) and mentions (mentioning another user).

A randomly selected portion of documents across all online genres were annotated for situational influence by seven annotators. The annotators were given the following general definition: An influencer is someone who has credibility in the group, persists in attempting to convince others, and introduces topics/ideas that others pick up on or support. In addition, the annotators were also provided with a detailed explanation of who is not an influencer as well as sample threads. Seven annotators annotated the Wikipedia and LiveJournal threads with an average IAA using Cohen's κ of 0.53 among all annotators on a portion of 12 Wikipedia Talk Pages. The political threads were annotated by four annotators with an average IAA

Corpus	Threads	Authors	Posts	Influencers	Threads w/o Influence
Wikipedia	509	3,130	8,749	456	76
LiveJournal	219	1,157	2,026	223	0
Political Forum	101	1,147	3,202	91	17
Create Debate	106	1,737	6,494	94	18
Twitter	99	1,602	1,919	101	4

Figure 21: List of Statistics for Each of the Annotated Influence Datasets

using Cohen’s κ of 0.57 among all annotators computed on six threads, two from each genre. Statistics on these datasets are presented in Figure 21.

We train a classifier to decide for each participant in a thread whether he or she is an influencer in that particular thread. We include threads from our dataset where there is at least one influencer, narrowing our task to finding the influencers when they exist. Since we know that there is at least one influencer in each discussion, we use the confidence of the classifier to rank the most likely influencer. We also use a supervised **domain adaptation** technique that augments the feature space among the target and source domains. The features and parameters used in our system were tuned using a development set comprising 10% each of the Wikipedia and LiveJournal datasets. Cross-validation was 10-fold for LiveJournal and Wikipedia, and 5-fold for Create Debate, Political Forum, and Twitter due to their small sizes. We also performed cross-domain experiments, by testing on a held out test set comprising of 10% each of the Wikipedia and LiveJournal datasets and 30% of the Create Debate, Political Forum, and Twitter datasets.

We compare our results to two baselines: predicting everyone as an influencer, and a classifier that uses the number of words a person wrote as its only feature. All following experiments include the number of words as a feature. Within each genre, we show the results using the features produced from each system component individually, all components (including with and without ranking), and the best components that differ per genre. In addition, we compare using only a single genre (e.g., training and testing on Wikipedia) and Domain Adaptation (e.g., Training on all genres, applying domain adaptation and testing on Wikipedia).

Cross-validation results are shown in Figure 22. The complete results of the experiments are shown in Figure 23. The last row shows the best results per experiment. The features used in the best system vary across genres in all experiments. We do see, however, that at least one of the dialectical features (i.e., argumentation, claim, opinion, or persuasion) appears in the best system for every genre with the exception of LiveJournal. LiveJournal is an anomaly in that using only dialog features results in the best performance. Other features that are often important across genres include dialog, credibility, and agreement.

12.2.4 Conclusions

We have created several system components that have been used to successfully detect influence in multiple online genres. In doing so, we have provided several valuable contributions to future research. The first contribution of our research is a rich set of datasets annotated for influence across several genres. We have been able to develop a rich resource consisting of over 1,000 discussions and 8,700 authors annotated for influence detection across five online genres. The second contribution of our research is a comprehensive cross-genre analysis of the system components in detecting influence and how it differs across genres. Knowing that different components work in different situations can be very useful. This information could be used to not only say who is influential but how to be influential. For example, showing a person which components they use to be influential could also help them improve how they influence others and help them adjust to different settings.

In the future, we would like to explore a semi-supervised approach to exploit unlabeled data as well as

	Wikipedia	LiveJournal	Twitter	Create Debate	Political Forum
All Yes	29.8%	31.6%	10.7%	11.7%	17.0%
Num Words	42.3%	60.9%	13.2%	18.9%	49.7%
Agreement	50.6%	73.5%	71.4 ^{aa} %	33.2%	56.0%
Argument	48.9%	61.4%	12.1%	30.6%	57.1%
Claim	48.1%	62.5%	8.0%	25.7%	49.1%
Credibility	42.1%	62.4%	67.8%	19.4%	48.2%
Author Trait	40.1%	58.8%	13.4%	17.8%	29.0%
Dialog	53.7%	75.6%	69.6%	27.6%	47.8%
Persuasion	43.7%	63.5%	12.3%	24.6%	45.2%
All	50.4%	66.1%	62.6%	27.3%	50.4%
All Ranking	50.4%	70.9%	62.7%	29.8%	55.0%
All Test	46.2%	65.2%	72.2%	26.0%	40.9%

Figure 22: Cross-Validation Results

using deep learning. We would also like to experiment with annotating the discussions by ranking all the participants in the discussion from most to least influential. We believe this would help reduce disagreement among annotators, because, although the annotators may not agree on the top influencer, they would agree on the top n influencers (though not necessarily in the same order). In addition, the influencer system detects who the influencers are in an individual thread. We would like to extend the system to track trends over time. Tracking a person over time to examine the trends that occur is extremely difficult for several reasons. It would require a considerable amount of annotation that is both timely and costly. Second, although the same people may post several times in a single corpus, it would be impossible to track the same person across online genres. Therefore, instead, we could track trends towards a particular topic over time, which will allow us to determine how a topic becomes influential.

12.3 Persuasive Influence

12.3.1 Introduction

Automatic detection of persuasion in online discussion is key to understanding how social media is used. A writer who is successful in changing the opinions of readers demonstrates *influence* over others and thus detecting persuasive posts that successfully change opinions is part of the overall solution to influence detection.

We conduct experiments on “Change My View”, a specific “sub-reddit” of the Reddit social media platform. “Change My View” (CMV) is a discussion forum where users post their opinions on a topic and their reasons for their beliefs. Other users respond by posting arguments attempting to change the view of the initiator of the discussion. If the views of the original posters are successfully changed, they will indicate this by posting a response with a “delta” character, providing naturally labeled data. Consider the example in Table 39. In this discussion, the Reddit user “A” states her belief and the user “B” responds with her own argument. The original poster “A” eventually responds with a delta and acknowledges that she doesn’t have a legitimate counter-argument.

In our experiments, we predict whether a user will reward another user with a “delta,” thus indicating the persuasive influence relationship. In other words, for each original poster X (of which each thread has exactly one), for all other users Y in the thread, we predict whether Y persuades X .

System	Wikipedia		LiveJournal		Political Forum		Create Debate		Twitter	
	SD	DA	SD	DA	SD	DA	SD	DA	SD	DA
All Yes	28.7	28.7	32.5	32.5	16.7	16.7	10.8	10.8	14.4	14.4
Num Words	41.0	39.1	56.7	32.3	37.5	40.5	22.4	22.1	16.5	0.0
Agreement	45.8 ^a	46.2 ^a	72.3 ^{aβ}	69.2 ^{aβ}	51.8 ^{aβ}	56.0 ^{aβ}	28.6 ^a	25.0 ^a	75.3 ^{aβ}	79.5 ^{aβ}
Argument	45.2 ^a	47.7 ^a	61.8 ^{aβ}	58.3 ^{aβ}	56.3 ^a	54.5 ^{aβ}	28.8 ^a	32.1 ^{aβ}	15.1	0.0
Claim	40.7 ^a	41.5 ^a	59.0 ^a	57.1 ^{aβ}	46.3 ^a	50.7 ^a	28.1 ^a	30.2 ^{aβ}	15.9	11.8
Credibility	36.5 ^a	31.3	60.0 ^a	47.8	37.5 ^a	41.9 ^a	17.7 ^a	23.2 ^a	73.0 ^{aβ}	73.0 ^{aβ}
Author Trait	37.2 ^a	31.0	52.3 ^a	36.0	25.0 ^a	25.6 ^a	15.9 ^a	27.8 ^{aγ}	13.5	9.8
Dialog	50.7 ^a	50.9 ^{aβ}	78.3 ^{aβ}	79.2 ^{aβ}	44.4 ^a	50.6 ^{aγ}	25.5 ^a	28.2 ^a	71.4 ^{aβ}	85.3 ^{aβγ}
Persuasion	40.7 ^a	40.8	59.3 ^a	47.8 ^a	48.8 ^a	53.8 ^{aβ}	26.9 ^a	27.5 ^a	16.5	6.1
All	46.2 ^a	51.2 ^{aβγ}	65.2 ^a	57.1	40.9 ^a	39.2 ^a	26.0 ^a	16.4	72.2 ^{aβ}	68.5 ^{aβ}
All Ranking	45.8 ^a	50.4 ^{aβγ}	65.3 ^a	58.3	44.4 ^a	43.3 ^a	25.3 ^a	14.5	74.0 ^{aβ}	69.3 ^{aβ}
Best	54.4 ^{aβ}	56.7 ^{aβ}	78.3 ^{aβ}	79.2 ^{aβ}	69.0 ^{aβ}	74.1 ^{aβ}	30.6 ^a	34.1 ^{aβ}	77.9 ^{aβ}	86.2 ^{aβ}

Figure 23: Influence Detection Results: The Results Are Shown when Training in a Single Domain (SD) and Applying Domain Adaptation (DA). Performance Is in Terms of F-score Using the Baselines (Everyone Is an Influencer, and Number of Words), All Features (Full), Individual Features One at a Time, and the Best Feature Combination for Each Data Set which Differs for each Genre. Significance Is Shown in Comparison to all-yes Baseline (α), Num-words Baseline (β), Domain Adaptation to Single Domain (γ)

12.3.2 Methods, Assumptions and Procedures

We model the posts using a hierarchical deep learning approach (Hidey and McKeown, 2018). Given a sentence representation $\mathbf{r}_s$ (see Section 12.3.2), where s is the index of the sentence, we model the document as a Long Short-Term Memory Network (LSTM) (Hochreiter and Schmidhuber, 1997a) with an attention mechanism over the sentences. We first apply a transformation to $\mathbf{r}_s$ to obtain a hidden state $\mathbf{h}_s$. Next, we compute a document representation with attention over each hidden state, similar to previous work:

$$\mathbf{h} = \text{SUM}_{s \in [1, S]} \alpha_s \mathbf{h}_s \quad (8)$$

where S is the number of sentences in the document and attention is calculated by applying an MLP to the hidden state, $\mathbf{u}_s = \tanh(W_s \mathbf{h}_s + \mathbf{b}_s)$, before calculating the probability distribution over sentences (using $\mathbf{q}$ as a learned parameter vector):

$$\alpha_s = (\exp(\mathbf{u}_s^T \mathbf{q}) / \text{SUM}_{i \in [1, S]} \exp(\mathbf{u}_i^T \mathbf{q})) \quad (9)$$

Finally, the document representation $\mathbf{h}$ is then passed through a multi-layer perceptron (MLP) to make a binary prediction of influence, which can be combined with features derived from the document:

$$y = \sigma(\text{MLP}(\mathbf{h}) + \beta^T \phi) \quad (10)$$

Sentence Representation. We create a sentence representation $\mathbf{r}_s$ by combining features from words, semantic frames, and discourse relations. We first represent each sentence by a weighted average of its word embeddings. Given a sentence at index s with T words and word embedding $\mathbf{x}^{\text{word}}_{s,t}$ for $t \in [1, T]$, the vector for s is:

$$\mathbf{v}_s^{\text{word}} = \text{SUM}_{t \in [1, T]} \alpha^{\text{word}}_{s,t} \mathbf{x}^{\text{word}}_{s,t} \quad (11)$$

Similarly, we add embeddings for semantic frames. The FrameNet model of frame semantics provides a method for describing events and relations. It also provides a way to model social interactions that are not

Table 39: Truncated Discussion Thread in Change My View

User	Post
Title	CMV: my view is that nations are just lines on a map and not real or useful
A	Nations are just lines on a map and don't exist in reality, here's my reasoning: 1) No one can decide where a nation begins or ends. Everyone's conception of "the South" when talking about America for example, will include different states and regions than the next person. In Europe, Turks claim that Cyprus is part of their nation, while Greece claims that island. Both claim Constantinople. Similarly, ...
B	https://en.wikipedia.org/wiki/Social_fact There is a word for what you are describing. While I'd concede your point is potentially valid, using your line of thinking makes living as a human being really difficult ... social facts make living in a human society possible in the first place. While they might be technically no true/real in a certain sense of the word, they provide structure in an otherwise structureless world. What's better? Have some orientationen, even though it's technically wrong. Or live without any kind of point of orientation, in a structureless world?
A	I'm going to give you a delta because you totally nailed it with the definition and your third paragraph raises points I can't answer: Δ

captured by discourse structure or explicitly expressed in words such as agreement and disagreement. For example, the verb “agree” may take the “Compatibility” frame, which is shared with similar verbs. We use a FrameNet parser to predict the labels for lexical units and represent frames as the weighted average of the labels:

$$\mathbf{v}_s^{frame} = \text{SUM}_{l \in [1, L]} \alpha_{s, l}^{frame} \mathbf{x}_{s, l}^{frame} \quad (12)$$

where $\mathbf{x}_{s, l}^{frame}$ is the embedding for the l^{th} frame and L is the total number of frames.

Each attention weight $\alpha_{s, j}^k$ is calculated for each $\mathbf{x}_{s, j}^k$ for k in $word, frame$ and J in $\{T, L\}$, respectively, where T is the number of words and L is the number of frames:

$$\alpha_{s, j}^k = (\exp(\mathbf{u}_{s, j}^{kT} \mathbf{q}_k) / \text{SUM}_{a \in [1, J]} \exp(\mathbf{u}_{s, a}^{kT} \mathbf{q}_k)) \quad (13)$$

and $\mathbf{u}_{s, j}^k = \tanh(W_k \mathbf{x}_{s, j}^k + \mathbf{b}_k)$ and $\mathbf{q}_k$ is a parameter vector.

Finally, we augment the sentence representation by incorporating embeddings for PDTB discourse structure. We represent the second-level discourse classes (e.g. Contingency/Causal and Comparison/Concession) for each inter-sentence relation as an embedding for sentence s as $\mathbf{v}_s^{inter}$, indicating the relationship between s and $s \pm$.

The final sentence representation is then determined by concatenating each component of the sentence :

$$\mathbf{v}_s = \mathbf{v}_s^{word}; \mathbf{v}_s^{frame}; \mathbf{v}_s^{inter} \quad (14)$$

Given $\mathbf{v}_s$, we could use this representation of each sentence for the input at each timestep of an LSTM, or model feature interaction by applying an MLP to $\mathbf{v}_s$. Instead, we allow the model to decide whether to carry features directly to the next layer, in order to allow for interaction between the word, discourse, and frame semantic features derived during this step. We thus obtain our sentence representation by feeding $\mathbf{v}_s$ into a highway network:

$$\mathbf{r}_s = \mathbf{t}_s \mathbf{z}_s + (1 - \mathbf{t}_s) \mathbf{v}_s \quad (15)$$

where $\mathbf{z}_s = g(W_h \mathbf{v}_s + \mathbf{b}_h)$, a hidden representation of the original vector with a non-linearity g , and $\mathbf{t}_s = \sigma(W_t \mathbf{v}_s + \mathbf{b}_t)$, a prediction of whether to use the original features. The highway network is a mixture of the hidden representation of the vector given by the MLP and the original vector, where the model learns the weight vector $\mathbf{t}_s$. Thus, because of the learned weight $\mathbf{t}_s$, the model decides how to interpolate between the hidden representation and the original vector.

Dynamic Memory Network. One variant of our model is to use a bi-directional LSTM for $\mathbf{h}_s$ in Equation 8 over the sentences from the reply only. However, this would only allow the attention mechanism to consider the response, rather than the context of the original post. We thus include information about the original post using a dynamic memory network to iteratively find abstract representations using information from both the original post and the response. Let $\mathbf{h}_s^r$ be the LSTM state at sentence s in the response and $\mathbf{h}_s^{op}$ the LSTM state at sentence s in the original post. We then create a representation $\mathbf{h}^{op}$ for the entire original post by using the attention mechanism in Equation 8 where $\mathbf{h}_s = \mathbf{h}_s^{op}$. This is concatenated with $\mathbf{h}_s^r$ and the memory representation $\mathbf{v}^t$ to create the input representation:

$$\mathbf{h}_s^t = [\mathbf{h}_s^r; \mathbf{h}_s^{op}; \mathbf{v}^t] \quad (16)$$

By allowing the attention mechanism to consider the context and the entire response, the model is able to more accurately predict which sentences are important. This results in a modified version of equation 8, where:

$$\mathbf{h}^t = \text{SUM}_{s \text{ in } [1, S]} \alpha_s \mathbf{h}_s^t \quad (17)$$

After each iteration, the memory $\mathbf{v}^t$ is set to $\mathbf{h}^{t-1}$. The initial memory $\mathbf{v}^0$ is initially set to the average of the hidden states in the response:

$$(1/S) \text{SUM}_{s \text{ in } [1, S]} \mathbf{h}_s^r \quad (18)$$

We could use $\mathbf{h}^0$ as the final document representation $\mathbf{h}$ in Equation 10, but in practice multiple iterations are more effective, which we validate empirically.

12.3.3 Results and Discussion

We use a dataset derived from the Change My View subreddit, a persuasive corpus where a user indicates if their view has been changed. As the data is self-labeled by posters, no human annotators are required. Given a post and response, we attempt to predict whether the user changed their view. For this task, we extract posts and automatically identify positive/negative examples as paths in a discussion tree terminating with/without a delta, respectively. We extract only one path per response to the original poster by following the left-most path in a depth-first search and allowing a single unique response per path. Each datapoint is then an original post and attempted persuasive response, where responses are one or more sequential posts from the same commenter. For training, we require every original post in the data to have at least 1 positive and 1 negative response. The resulting training set has 19516 examples (14849 negative and 4667 positive). The test set contains 2465 examples (1836 negative and 629 positive). We distinguish two cases of the path-based prediction: predicting a delta from only the initial post in the response (termed the *root reply*) and including all posts in the response (termed the *full path*). For our experiments, at minimum the root reply and/or original post are available.

The results of our experiments on the held-out test set are shown in Table 40. We report results for both the root reply and full path options and we compare models using sentences from just the response (R) and the response plus the original post (R+OP). We provide baseline models, from previous work, trained using logistic regression on features from just the response (*bag-of-words*) and from the response plus the original post (*interplay*). In the work of Tan et al., their best-performing features were derived from the *interplay* (IP) between the original post and the response. They derived 12 features from 4 similarity scores (common words, similar fraction in reply, similar fraction in OP, and Jaccard score) and 3 subsets (all words, stop words, and content words). The interplay provides a strong baseline because we might expect there to be significant overlap between the posts if users are imitating the writing style of the original poster in order to be more persuasive. In addition, we provide a *bag-of-words* (BoW) baseline. We remove words occurring less than 5 times and L2-normalize term frequency vectors. We present results using only words (word-LSTM) and words, frames, and discourse relations (all-LSTM). These models consider the response

only. When the response and original post are both provided, we use the memory network described in 12.3.2 (all-LSTM+memory). We also provide results for baseline features combined with our model (all-LSTM+memory+IP), with the features as φ in Equation 10.

Table 40: Results of Persuasive Influence Prediction Task

	Model	Root Reply			Full Path		
		Acc.	AUC	True F-score	Acc.	AUC	True F-score
R	BoW	60.4	68.9	47.1	61.9	72.8	50.3
	word-LSTM	71.2	70.5	48.7	72.9	75.1	52.7
	all-LSTM	72.5	70.8	48.9	75.1	75.5	53.0
R+OP	IP	70.5	74.8	52.1	72.7	76.7	54.6
	all-LSTM+memory	75.0	74.9	53.1	74.3	77.3	55.4
	all-LSTM+memory+IP	77.2	79.5	58.0	81.0	82.1	60.7

12.3.4 Conclusions

We have presented evidence that the ordering of a document is crucial to influential writing. We provided a neural model using words, frames, and discourse relations that effectively predicts persuasiveness in several tasks and significantly improves upon prior work.

In future work, we hope to continue work in influence and persuasiveness. This dataset has the advantage of being labeled, but work in *unsupervised* persuasiveness prediction, given only text responses indicating persuasiveness, is one possible direction. Other avenues of research include modeling the interaction between the original post and the responses. Interplay is a simple but effective representation of interaction but modeling threads as dialogues or multi-party discourse rather than monologues may yield further improvements. Finally, the users in Change My View are required to provide an explanation for the reason their view changed and we can analyze these reasons and attempt to predict *why* someone changed their view.

Chapter 13

Power Relations

13.1 Summary

13.1.1 Goal of Work

The nature of the relationship between discourse participants affects how they communicate. While this fact is of interest to sociolinguists, this means that in a computational setting we can determine relationships between discourse participants based on how they communicate. We have studied the relationship of institutional power, by which we mean the fact that by virtue of belonging to an organization two people are in a hierarchical relationship.

13.1.2 Main Methods

The main approach relies on supervised machine learning using support vector machines. We use data from the Enron corpus of emails; we have augmented this corpus with the organizational hierarchy so we have a gold standard for training and testing. While email has not been the focus of the DARPA DEFT project, no other comparable data is available. It is clear that power relations would affect communication in a large variety of interaction modes.

13.1.3 Resulting Resources

We have released a version of the Enron corpus augmented with the following information:

- The organizational hierarchy which we derived from charts found as attachments in the email.
- Information about the gender of senders and receivers.
- Some annotations on power as determined by annotators. This covers different types of power (influence, situational power, and hierarchical power).

This version of the corpus is available at <https://sites.google.com/site/columbiawisrlab/resources>.

13.2 Introduction

Computationally analyzing the social context in which language is used has gathered great interest within the NLP community recently. One of the areas that has generated substantial research is the study of how social power relations between people affect and/or are revealed in their interactions with one another. Researchers have proposed systems to detect social power relations between participants of organizational email threads (Bramsen et al., 2011; Gilbert, 2012; Prabhakaran and Rambow, 2013b), online forums (Danescu-Niculescu-Mizil et al., 2012; Biran et al., 2012; Danescu-Niculescu-Mizil et al., 2013), chats (Strzalkowski et al., 2012),

and off-line interactions such as presidential debates (Prabhakaran et al., 2013; Nguyen et al., 2013). Automatically identifying power and influence from interactions can have many practical applications ranging from law enforcement and intelligence to online marketing.

A significant number of these studies are performed in the domain of organizational email where there is a well defined notion of power (organizational hierarchy). Bramsen et al. (2011) and Gilbert (2012) predict hierarchical power relations between people in the Enron email corpus using lexical features extracted from *all* the messages exchanged between them. However, their approaches primarily apply to situations where large collections of messages exchanged between pairs of people are available. In (Prabhakaran and Rambow, 2013b), we introduced the problem of detecting whether a participant of an email thread has power over someone else in the thread and established the importance of dialog structure in that task. However, in that work we did not detect over whom that person has power.

In this paper, we introduce a new problem formulation. We predict the hierarchical power relation between pairs of participants in an email interaction thread based *solely* on features extracted from that thread. As a second major contribution, we introduce a new set of features to capture aspects of participant behavior such as responsiveness, and we show that these features are significantly correlated with the direction of power. We present a fully automatic system for this task obtaining an accuracy of 73.0%, an improvement of 6.9% over 68.3% by a system using only lexical features. This best-performing system uses our new feature set.

13.3 Methods, Assumptions, and Procedures

13.3.1 Motivation

Early NLP-based approaches such as Bramsen et al. (2011) and Gilbert (2012) built systems to predict hierarchical power relations between people in the Enron email corpus using lexical features from all the messages exchanged between them. One limitation of this approach is that it relies solely on lexical cues and hence works best when large collections of messages exchanged between the pairs of people are available. For example, Bramsen et al. (2011) excluded sender-recipient pairs who exchanged fewer than 500 words from their evaluation set, since they found smaller text samples are harder to classify. By taking the message out of the context of the interaction in which it was exchanged, they fail to utilize cues from the structure of interactions, which complements the lexical cues in detecting power relations, as we showed in (Prabhakaran and Rambow, 2013b).

We modeled the problem of detecting power relationships differently in (Prabhakaran and Rambow, 2013b): we predicted whether a participant in an email thread has a certain type of power or not. However, in that work we did not predict over whom he/she has that power. This may result in noisy features; consider a thread in which participant *X* has power over participant *Y*, who has power over participant *Z*. By aggregating features over all messages sent by *Y*, features salient to a subordinate-superior interaction are incorrectly conflated with those salient to superior-subordinate interaction. Another limitation of (Prabhakaran and Rambow, 2013b) is that we used manual annotations for many of our features such as dialog acts and overt displays of power. Relying on manual annotations for features limited our analysis to a small subset of the Enron corpus, which has only 18 instances of hierarchical power. Consequently, our findings with respect to hierarchical power were weak in terms of both correlations of features and system performance.

In this paper, we introduce the problem of predicting who has power over whom in pairs of interacting participants based on a single thread of interactions. From (Bramsen et al., 2011) we retain the idea that we want to predict the power relation between pairs of people. But in contrast to their formulation, we retain the goal from (Prabhakaran and Rambow, 2013b) that we want to study communication in the context of an interaction, and that we want to be able to make predictions using only the emails exchanged in a single thread. Like (Prabhakaran and Rambow, 2013b), we use features to capture the dialog structure, but we use automatic taggers to generate them and assume no manual annotation at all at training or test time. This

allows us to use the entire Enron email corpus for this study.

13.3.2 Data

In this work, we use the version of Enron email corpus by Yeh and Harnly (2006) which captures the thread structure of email exchanges. The corpus contains 36,615 email threads. We excluded a small subset of 419 threads that was used for previous manual annotation efforts, part of which was also used to train the DA and ODP taggers (Section 13.3.4) that generate features for our system. The average number of email messages per thread was around 3. We divided the remaining threads into train (50%), dev (25%) and test (25%) sets by random sampling. We then applied various basic NLP preprocessing steps such as tokenization, POS tagging and lemmatization to the body of email messages. We use the Enron gold organizational hierarchy released by Agarwal et al. (2012b) to model hierarchical power. Their corpus was manually built using information from Enron organizational charts. It includes relations of 1,518 employees and captures dominance relations between 13,724 pairs of them. This is the largest such data set available to the best of our knowledge.

13.3.3 Problem Formulation

Let t denote an email thread and M_t denote the set of all messages in t . Also, let P_t be the set of all participants in t , i.e., the union of senders and recipients (*To* and *CC*) of all messages in M_t . We are interested in detecting power relations between pairs of participants who interact within a given email thread. Not every pair of participants (p_1, p_2) in $P_t \times P_t$ interact with one another within t . Let $IM_t(p_1, p_2)$ denote the set of *Interaction Messages* — non-empty messages in t in which either p_1 is the sender and p_2 is one of the recipients or vice versa. We call the set of (p_1, p_2) such that $|IM_t(p_1, p_2)| > 0$ the *interacting participant pairs* of t (IPP_t). We focus on the manifestations of power in interactions between people across different levels of hierarchy. For every (p_1, p_2) in IPP_t , we query the set of dominance relations in the gold hierarchy to determine their hierarchical power relation ($HP(p_1, p_2)$). We exclude pairs that do not exist in the gold hierarchy from our analysis and denote the remaining set of *related interacting participant pairs* as $RIPP_t$. We assign $HP(p_1, p_2)$ to be *superior* if p_1 dominates p_2 , and *subordinate* if p_2 dominates p_1 . Table 41 shows the total number of pairs in IPP_t and $RIPP_t$ from all the threads in our corpus and across *train*, *dev* and *test* sets.

Table 41: Data Statistics. Row 1 presents the total number of threads in different subsets of the corpus. Row 2 and 3 present the number of interacting participant pairs (IPP) and related interacting participant pairs ($RIPP$) in those subsets.

Description	Total	Train	Dev	Test
# of threads	36,196	18,079	8,973	9,144
$\sum_t IPP_t $	355,797	174,892	91,898	89,007
$\sum_t RIPP_t $	15,048	7,510	3,578	3,960

Given a thread t and a pair of participants (p_1, p_2) in $RIPP_t$, we want to automatically detect $HP(p_1, p_2)$. This problem formulation is similar to the ones in (Bramsen et al., 2011) and (Gilbert, 2012). However, the difference is that for us an instance is a pair of participants in a single thread of interaction (which may or may not include other people), whereas for them an instance constitutes all messages exchanged between a pair of people in the entire corpus. Our formulation also differs from (Prabhakaran and Rambow, 2013b) in that we detect power relations between pairs of participants, instead of just whether a participant had power over anyone in the thread.

13.3.4 Structural Analysis

In this section we analyze various features that capture the structure of interaction between the pairs of participants in a thread. Each feature f is extracted with respect to a person p over a reference set of messages M (denoted f^p_M). For a pair (p_1, p_2) , we extract 4 versions of each feature f : $f^{p_1}_{IMt(p_1, p_2)}$, $f^{p_2}_{IMt(p_1, p_2)}$, $f^{p_1}_{Mt}$ and $f^{p_2}_{Mt}$. The first two capture behavior of the pair among themselves, while the third and fourth capture their overall behavior in the entire thread. We group our features into three categories — THR^{New} , THR^{PR} and DIA^{PR} . THR^{New} is a set of new features we propose, while THR^{PR} and DIA^{PR} incorporate features we proposed in (Prabhakaran and Rambow, 2013b). THR^{New} and THR^{PR} capture the structure of message exchanges without looking at the content of the emails (e.g., how many emails did a person send), while DIA^{PR} captures the pragmatics of the dialog and requires an analysis of the content of the emails (e.g., did they issue any requests).

THR^{New} : This is a new set of features we introduce in this paper. It includes the average number of recipients (AvgRecipients) and *To* recipients (AvgToRecipients) in emails sent by p , the percentage of emails p received in which he/she was in the *To* list (InToList%), boolean features denoting whether p added or removed people when responding to a message (AddPerson and RemovePerson), average number of replies received per message sent by p (ReplyRate) and average number of replies received from the other person of the pair to messages where he/she was a *To* recipient (ReplyRateWithinPair). ReplyRateWithinPair applies only to $IMt(p_1, p_2)$.

THR^{PR} : This feature set includes two meta-data based feature sets — positional and verbosity. Positional features include a boolean feature to denote whether p sent the first message (Initiate), and relative positions of p 's first and last messages (FirstMsgPos and LastMsgPos) in M . Verbosity features include p 's message count (MsgCount), message ratio (MsgRatio), token count (TokenCount), token ratio (TokenRatio) and tokens per message (TokenPerMsg), all calculated over M .

DIA^{PR} : In (Prabhakaran and Rambow, 2013b), we used dialog features derived from manual annotations — dialog acts (DA) and overt displays of power (ODP) — to model the structure of interactions within the message content. In this work, we obtain DA and ODP tags on the entire corpus using automatic taggers trained on those manual annotations. The DA tagger (Omuya et al., 2013) obtained an accuracy of 92%. The ODP tagger (Prabhakaran et al., 2012b) obtained an accuracy of 96% and F-measure of 54%. The DA tagger labels each sentence to be one of the 4 dialog acts: Request Action, Request Information, Inform, and Conventional. The ODP Tagger identifies sentences (mostly requests) that express additional constraints on its response, beyond those introduced by the dialog act. We use 5 features: ReqAction%, ReqInform%, Inform%, Conventional%, and ODP% to capture the percentage of sentences in messages sent by p that has each of these labels. We also use a feature to capture the number of p 's messages with a request that did not get a reply, i.e., dangling requests (DanglingReq%), over all messages sent by p .

We perform an unpaired two-sample two-tailed Student's t-Test comparing mean values of each feature for *subordinates* vs. *superiors*. For our analysis, a data point is a related interacting pair, and not a message. Hence, a message with multiple recipients who have a *superior/subordinate* relation with the sender will contribute to features for multiple data points. We limit our analysis to the related interacting pairs from only our *train* set. Table 42 presents mean values of features for *subordinates* and *superiors* at the interaction level. Thread level versions of these features also obtained similar results overall in terms of direction of difference and significance. We denote three significance levels — * ($p < .05$), ** ($p < .01$), and *** ($p < .001$). To control false discovery rates in multiple testing, we adjusted the p-values (Benjamini and Hochberg, 1995). We summarize the main findings on the significant features below.

1. Superiors send messages addressed to more people (AvgRecipients and AvgToRecipients). Consequently, they get more replies to their messages (ReplyRate). However, considering messages where the other person of the pair is addressed in the *To* list (ReplyRateWithinPair), subordinates get more replies.

Table 42: Student's t-Test Results of f^{p}_{IMt} . THR^{New} : new meta-data features; THR^{PR} , DIA^{PR} : meta-data and dialog-act features from previous studies;

* ($p < .05$); ** ($p < .01$); *** ($p < .001$).

Feature Name	Mean(f^{sub}_{IMt})	Mean(f^{sup}_{IMt})
<i>THR^{New}</i>		
AvgRecipients ***	21.14	43.10
AvgToRecipients ***	18.19	38.94
InToList%	0.82	0.80
ReplyRate ***	0.86	1.23
ReplyRateWithinPair ***	0.16	0.10
AddPerson	0.48	0.47
RemovePerson ***	0.41	0.37
<i>THR^{PR}</i>		
Initiate ***	0.45	0.56
FirstMsgPos	0.04	0.03
LastMsgPos ***	0.15	0.11
MsgCount ***	0.64	0.70
MsgRatio ***	0.44	0.56
TokenCount	91.22	83.26
TokenRatio ***	0.45	0.55
TokenPerMsg *	140.60	120.87
<i>DIA^{PR}</i>		
Conventional% ***	0.15	0.17
Inform% ***	0.78	0.72
ReqAction% ***	0.02	0.04
ReqInform% ***	0.05	0.06
DanglingReq% ***	0.12	0.15
ODP% ***	0.03	0.06

2. Superiors issue more requests for action (ReqAction%) and overt displays of power (ODP%), while subordinates issue more informs (Inform%) and more of their requests went unanswered (DanglingReq%).
3. Superiors initiate the interactions more often than subordinates (Initiate). They also leave interactions earlier (LastMsgPos).
4. Superiors send shorter messages (TokenPerMsg). They also send more messages (MsgCount & MsgRatio) and even contribute a higher ratio of tokens in the thread (TokenRatio) despite sending shorter messages.

Finding 1 goes in line with findings from studies analyzing social networks that superiors have higher connectivity in the networks that they are part of (Rowe et al., 2007b). Intuitively, those who have higher connectivity also send emails to a larger number of people, and hence our result. Since superiors address more people in their emails, they also have a higher chance of getting replies. Finding 2 also aligns with the general intuition about how superiors and subordinates behave within interactions (e.g., superiors exhibit more overt displays of power than subordinates).

Findings 3 & 4 are interesting since they reveal special characteristics of threads involving hierarchically related participants. In (Prabhakaran and Rambow, 2013b), we had found that persons with hierarchical

power rarely initiated threads and contributed less within the threads. But that problem formulation was different — we were identifying whether a person in a given thread had hierarchical power over someone else or not. The data points in that formulation included participants from threads that did not have any hierarchically related people, whereas our current formulation does not. These findings suggest that if a person starts an email thread, he’s likely not to be the one who has power, but if a thread includes a pair of people who are hierarchically related, then it is likely to be initiated by the superior and he/she tends to contribute more in such threads.

13.4 Results and Discussion

We build an SVM-based supervised learning system that can predict $HP(p_1, p_2)$ to be either *superior* or *subordinate* based on the interaction within a thread t for any pair of participants (p_1, p_2) in $RIPP\ t$. We deterministically fix the order of participants in (p_1, p_2) such that p_1 is the sender of the first message in $IM\ t(p_1, p_2)$. We use the ClearTK (Ogren et al., 2008) wrapper for SVMLight (Joachims, 1999) in our experiments. We use the related interacting participant pairs in threads from the *train* set to train our models and optimize our performance on those from the *dev* set. We report results obtained on *dev* and *test* sets.

In our formulation, values of many features are undefined for some instances (e.g., Inform% is undefined when MsgCount = 0). Handling of undefined values for features in SVM is not straightforward. Most SVM implementations assume the value of 0 by default in such cases, conflating them with cases where Inform% is truly 0. In order to mitigate this issue, we use an indicator feature for each structural feature to denote whether or not it is valid. Since we use a quadratic kernel, we expect the SVM to pick up the interaction between each feature and its indicator feature.

Lexical features have already been shown to be valuable in predicting power relations (Bramsen et al., 2011; Gilbert, 2012). We use another feature set LEX to capture word ngrams, POS (part of speech) ngrams and mixed ngrams. A mixed ngram (Prabhakaran et al., 2012b) is a special case of word ngram where words belonging to open classes are replaced with their POS tags. We found the best setting to be using both unigrams and bigrams for all three types of ngrams, by tuning in our *dev* set. We then performed experiments using all subsets of $\{LEX, THR^{New}, THR^{PR}, DIA^{PR}\}$.

Table 43 presents the results obtained using various feature subsets. We use a majority class baseline assigning $HP(p_1, p_2)$ to be always *superior*, which obtains 52.5% accuracy. We also use a stronger baseline using word unigrams and bigrams as features, which obtained an accuracy of 68.6%. The performance of the system using each structural feature class on its own is very low. Combining all three of them improves the accuracy to 61.8%. The highest performance obtained without using any message content is for THR^{PR} and THR^{New} (61.5%). LEX features by itself obtain a very high accuracy of 70.7%, confirming the importance of lexical patterns in this task. Perplexingly, adding all structural features to LEX reduces the accuracy by around 2.3 percentage points. The best performing system (BEST) uses LEX and THR^{New} features and obtains an accuracy of 73.0%, a statistically significant improvement over the LEX-only system (McNemar).

We also performed an ablation study to understand the importance of different slices of our feature sets. If we remove all feature versions with respect to the second person, the accuracy drops to 72.1%. This suggests that features about the other person’s behavior also help the prediction task. If we remove either the thread level versions of features or interaction level versions of features, the accuracy again drops, suggesting that both the pair’s behavior among themselves, and their overall behavior in the thread add value to the prediction task. Removing the indicator feature denoting the structural features’ validity also reduces the performance of the system.

We now discuss evaluation on our blind test set. The majority baseline (Always Superior) for accuracy is 55.0%. The word unigrams and bigrams baseline obtains an accuracy of 68.3%. The LEX system (using other forms of ngrams as well) obtains a slightly lower accuracy of 68.1%. Our BEST system using LEX and THR^{New} features obtains an accuracy of 73.0% (coincidentally the same as on the *dev* set), an

Table 43: Accuracies on feature subsets (*dev* set). THR^{New} : new meta-data features; THR^{PR} , DIA^{PR} : meta-data and dialog-act features from previous studies; LEX: ngrams; BEST: best subset; IMt stands for IMt(p_1 , p_2).

Description	Accuracy
Baseline (Always Superior)	52.54
Baseline (Word Unigrams + Bigrams)	68.56
THR^{New}	55.90
THR^{PR}	54.30
DIA^{PR}	54.05
$\text{THR}^{\text{PR}} + \text{THR}^{\text{New}}$	61.49
$\text{DIA}^{\text{PR}} + \text{THR}^{\text{PR}} + \text{THR}^{\text{New}}$	62.47
LEX	70.74
LEX + $\text{DIA}^{\text{PR}} + \text{THR}^{\text{PR}}$	67.44
LEX + $\text{DIA}^{\text{PR}} + \text{THR}^{\text{PR}} + \text{THR}^{\text{New}}$	68.56
BEST (= LEX + THR^{New})	73.03
BEST (Using p_1 features only)	72.08
BEST (Using <i>IM</i> t features only)	72.11
BEST (Using <i>M</i> t only)	71.27
BEST (No Indicator Variables)	72.44

improvement of 6.9% over the system using only lexical features.

13.5 Conclusions

We introduced the problem of predicting who has power over whom based on a single thread of written interactions. We introduced a new set of features which describe the structure of the dialog. Using this feature set, we obtain an accuracy of 73.0% on a blind test. In future work, we will tackle the problem of three-way classification of pairs of participants, which will cover cases in which they are not in a power relation at all.

Chapter 14

Social Network Extraction

14.1 Summary

14.1.1 Goal of Work

Data that is derived from electronic communication can be used directly to induce a social network, for example email records (who sent mail to whom?) or platforms such as Twitter (who follows whom? who retweets or answers whom?) or FaceBook (who is friends with whom? who chats with whom? who likes whose posts?). However, a lot of information about social relations is included in narrative texts: newspaper articles, spoken news transcripts, reports, and so on. The goal of this line of work is to extract a social network from narrative text. The input is simple text, not communication with meta data.

In this chapter, we discuss a case study in detail: the extraction of a social network from *Alice in Wonderland*. We use literature since it is readily accessible and describes interesting interactions among many recurring characters. In contrast, individual newspaper stories cover a relatively small number of people with limited interactions.

14.1.2 Resulting Resources

We have created an annotated corpus (which is available). The system is complex and relies on several components and is not available.

14.2 Introduction

Social network analysis affects a wide range of academic disciplines and practical applications: psychology (Seidman, 1985; Koehly and Shivy, 1998), anthropology (Sanjek, 1974; Johnson, 1994; Hage and Harary, 1983), political science (Knoke, 1990; Brandes et al., 2001), literary theory (Moretti, 2005), management (Tichy et al., 1979; Cross et al., 2001; Borgatti and Cross, 2003), and crime prevention and intelligence (Sparrow, 1991). In the past, social networks were constructed through interviews, surveys and experiments.

With the advent of the internet and online social networks, researchers have started constructing networks using meta-data that reflects interactions, such as self-declared friendship linkages, sender-receiver email linkages, comments on a common blog-post, etc. However, these methodologies of creating social networks ignore a vastly rich network expressed in the unstructured text of such sources. Moreover, many rich sources of social networks remain untouched simply because there is no meta-data associated with them (literary texts, news stories, historical texts). There have been recent efforts to extract social networks from text by mining interactions between people expressed linguistically in unstructured text (Elson et al., 2010; He et al., 2013). However, these approaches are restricted to extracting interactions signaled by quoted speech.

The chapter is structured as follows: In section 14.3.1 we describe our notion of social events and the annotated data we use for training and testing in our experiments. In section 14.3.2, we briefly describe the tree kernel structures used by our system to detect social events in text. Section 14.3.3 presents the social

network analysis metrics we use to evaluate the quality of the predicted network. In section 14.4, we present the experiments and results. We discuss some related work in section 14.3.4.

14.3 Methods, Assumptions and Procedures

14.3.1 Social Events and Data

In Agarwal et al. (2010), we defined a **social event** as an event in which two people *interact* such that for at least one person, the interaction is **deliberate** or **conscious**. Put differently, at least one person must be aware of the interaction.

[Toujan Faisal], 54,{ said} [she] was { informed} of the refusal by an [Interior Ministry committee] overseeing election preparations.

In the above example, the people (or groups of people) involved in social events are *Toujan Faisal* and the *Interior Ministry committee*. There are two social events in this example: one described by the word *said*, in which *Toujan Faisal* is *talking about* the committee, and the other described by the word *informed*, in which *Toujan Faisal* presumably has a mutual interaction with the committee.

We annotated two corpora for social events: 1) The Automatic Content Extraction (ACE) data-set¹ (Agarwal et al., 2010) and 2) the *Alice in Wonderland* data-set² (Agarwal et al., 2012a).

For each pair of entity mentions in a sentence, if the annotators annotate a social event, we count this as a positive example for the task of social event detection. If no social event is annotated between a pair of entity mentions, we count this as a negative example. Note that we only consider pairs of entity mentions that correspond to different entities; our annotation scheme disregards self-interactions (talking to oneself).

We use all of ACE data for training and refer to this data-set as **ACE-train**. We use all of *Alice in Wonderland* data for testing and refer to this data-set as **Alice-test**. The distribution of these data-sets is presented in Table 44.

Table 44: The distribution of social events in the training and test sets used for experiments

Data-set	# of social events	# of No-event
ACE-train	396	1,101
Alice-test	81	128

14.3.2 SINNET: Social Interaction Network Extractor from Text

In Agarwal and Rambow (2010a), we presented a preliminary system that extracts social events from news articles. We used Support Vector Machines (SVM) in conjunction with tree kernels for detecting social events between pairs of entities, called target entities, that co-occur in a sentence. Following is a brief description of the tree structures that we used for building our models. We used the Stanford parser's (Klein and Manning, 2003) phrase structure and dependency tree representations. Of the following tree structures, 1-3 have previously been proposed by Nguyen et al. (2009) for the relation extraction task, while we introduced the fourth structure in Agarwal and Rambow (2010a) for social event detection task.

1. PET: This refers to the smallest phrase structure tree that contains the two target entities.
2. Grammatical Relation (GR) tree: This refers to the smallest dependency tree that contains the two target entities. We replace the words (in the nodes of the tree) with their corresponding grammatical

¹Version: 6.0, Catalog number: LDC2005E18

²<http://www.gutenberg.org/ebooks/19551>

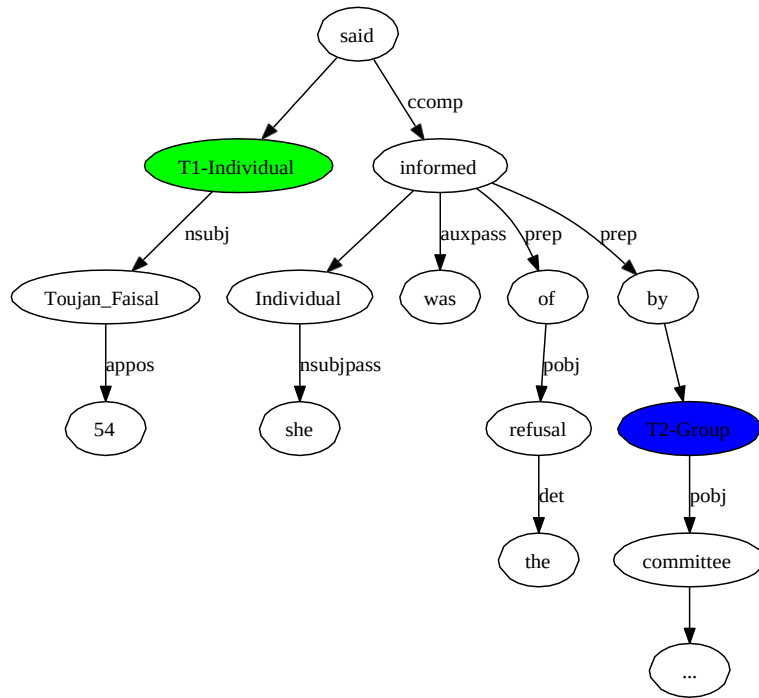


Figure 24: Dependency parse tree for the sentence (in the ACE corpus): *[Toujan Faisal], 54, said [she] was informed of the refusal by an [Interior Ministry committee] overseeing election preparations.*

roles. For example, in Figure 24, if we replace *Toujan_Faisal* by *nsubj*, *54* by *appos*, *she* by *nsubjpass* and so on, we will get a GR tree.

3. Grammatical Relation Word (GRW) tree: We get this tree by adding the grammatical relations as separate nodes between a node and its parent. For example, in Figure 24, if we add *nsubj* as a node between *T1-Individual* and *Toujan_Faisal*, add *appos* as a node between *54* and *Toujan_Faisal*, and so on, we will get a GRW tree.
4. Sequence in GRW tree (SqGRW): This is the sequence of nodes from one target to the other in the GRW tree. For example, in Figure 24, this would be *Toujan_Faisal nsubj T1-Individual said ccomp informed prep by T2-Group pobj committee*.

We also use combinations of the aforementioned structures. For example, *PET_GR_SqGRW* refers to a kernel that considers a linear combination of three structures (PET, GR and SqGRW) for calculating similarities between examples. We use the Partial Tree kernel, first proposed by Moschitti (2006a), to calculate similarities between these tree structures.

In this chapter, we use a Bag of Words Model (BOW) as a baseline. In the BOW model, each sentence is represented as a vector of three feature spaces. The first feature space encodes the presence and absence of words between the start of the sentence and the start of the first target entity mention. The second feature space encodes the presence and absence of words between the end of the first target entity mention and the start of the second target entity mention. The third feature space encodes the presence and absence of words between the end of the second target entity mention and the end of the sentence. This feature space has

previously been used by GuoDong et al. (2005) for the relation extraction task on ACE. We use stemming and remove stop words from our feature space.

14.3.3 Social Network Analysis Metrics

In this section we briefly discuss some of the most popular social network analysis (SNA) metrics used by researchers to mine information from networks. We evaluate the social network extracted by our system with the gold network using these metrics. At a broad level, SNA researchers are interested in measuring importance of nodes in the network and in finding community structures in the network. To measure the importance of nodes, they use the notion of centrality. Following are the centrality measures that are often used in the literature (Freeman, 1979):

1. **Degree centrality** of a node in the network measures the number of incoming and outgoing links from the node. Degree centrality is viewed as an index of a node's *communication activity*.
2. **Betweenness centrality** of a node in the network measures the frequency with which a point falls between pairs of other nodes on the shortest paths connecting them. Nodes with high betweenness centrality are strategically located on the communication paths linking pairs of others, thus having the potential of influencing the group by withholding or distorting information (Bavelas, 1948; Shaw, 1954; Shimmel, 1953).

Another aspect of social networks that SNA researchers are interested in has to do with finding communities in the network and structural properties of networks. Following are some basic metrics used for this task:

1. **Graph density**: The density of a graph is the ratio of the number of edges to the number of possible edges. This measures how close the network is to being complete.
2. **Connected components**: a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by some path. The number of connected components is an indication of the overall connectivity of the network.
3. **Triads**: A triad is a set of three parties connected pair-wise to each other. In his seminal work, Simmel (1950) argued that triads are a fundamental unit of sociological analysis. He argued that three actors in a triad may allow qualitatively different social dynamics that cannot be reduced to individuals or dyads.

14.3.4 Literature Survey

With the advent of the internet and social media, researchers have got access to different forms of communication such as Email (Klimt and Yang, 2004; Rowe et al., 2007a), online discussion threads (Hassan et al., 2012), Slashdot, Epinions, and Wikipedia (Leskovec et al., 2010). There have also been approaches of extracting networks based on Information Retrieval techniques – Jing et al. (2007) extract a network from conversational speech data. The events they are interested in are custody, death, hiding, liberation, marriage, migration, survival and violence. Tang et al. (2008) aim at extracting and mining academic social networks. Culotta et al. (2004) extract social networks and contact information from email and the Web. Mori et al. (2006) mine networks based on the collective context in which entities appear.

Our notion of social network is different from the aforementioned work. We are interested in extracting *interaction* networks from unstructured text. In terms of our goals, our work is closest to the work by Elson et al. (2010) and He et al. (2013). Elson et al. (2010) and He et al. (2013) are also interested in extracting a social network from literary texts. However, they restrict their definition of *interaction* to interactions that are signaled by quotation marks. Our system does not have this limitation and is therefore able to extract interaction links appearing even in reported speech (non-dialogue text).

14.4 Results and Discussion

We present experiments and results for two tasks: social event detection and social network extraction. We use the same system for both tasks; the first task is an intrinsic evaluation of our system, while the second task presents an extrinsic evaluation of our system. In the following subsections, we describe the individual tasks, their experimental set-up followed by a discussion of results.

14.4.1 Social Event Detection

Task description: Given a pair of entity mentions in a sentence, we evaluate how well we identify the occurrence of a social event between these two entities. This is a binary task with two classes: presence/absence of a social event. We evaluate using F-measure on the presence of social events.

Table 45: Results for training on ACE-train and testing on Alice-test. P refers to Precision, R refers to Recall, F refers to F1-measure. Terminology used for the tree structures is explained in detail in Section 14.3.2.

Tree structure	P	R	F
BOW	34.62	77.78	47.91
PET	58.54	59.26	58.90
GRW	49.14	70.37	57.87
SqGRW	49.59	74.07	59.41
PET_GR	56.32	60.49	58.33
PET_GR_SqGRW	56.82	61.73	59.17
GR_SqGRW	54.37	69.14	60.87
GRW_SqGRW	51.30	72.84	60.20
GR_GRW_SqGRW	50.47	66.67	57.45

Experimental set-up: For all our experiments, we use the SVM-Light-TK package (Moschitti, 2006b) for training models. We use the default parameters of the package to avoid over-fitting. Since we are interested in knowing how well we do at finding the social events, we report Precision, Recall and F-measure of the class of interest (the minority class) instead of % accuracy. For training, we set the γ parameter of the package to the ratio of the number of negative examples to the number of positive examples in the training data-set. The β parameter assigns a weight to the minority class. Since the SVM optimizes for accuracy, if we do not set this parameter at the time of training, the learner may learn a trivial hyperplane classifying all the examples as negative (the majority class), thus achieving a high accuracy. By assigning a weight to the examples in the minority class, we increase the cost of mis-classifying these examples, thus forcing the learner to find a non-trivial hyperplane. We use the model trained just on bag-of-words (BOW) as a baseline. Discussion of results: Table 45 presents the results for models trained on ACE-train and tested on Alice-test. We use the tree kernel structure combinations described in section 14.3.2. The results show that building a model using tree kernels outperforms the bag-of-words baseline model by an absolute 10% F1-measure. This difference is statistically significant with $p < 0.05$.

14.4.2 Social Network Extraction

In this section, we provide results to establish that the un-weighted network extracted using social event detection models is *close* to the true underlying network.

Task description and experimental set-up: Using our social event detection models, we build an un-weighted network of entities in *Alice in Wonderland*. For this task, we report the *distance* between the SNA metrics in the predicted and gold networks. Table 46 summarizes the metrics we use for our evaluation and elaborates on the meaning of *distance*. We use two baseline systems for our evaluation:

Table 46: Terminology used to present results in Table 47

Symbol	Name and explanation
P, R, F, %A	Precision, Recall, F1-measure and Accuracy
p	McNemar’s two-sided p-value significance test. We linearize the adjacency matrix of the predicted and gold network and test if these two vectors are significantly different.
D, B	Degree and Betweenness centrality. For each of these centrality measures, we find the centrality of nodes in the network, represented by a vector $\mathbf{v}$, where the i^{th} component of the vector is the centrality of the i^{th} node. We then calculate the Euclidean distance between the predicted and gold vectors, which measures the difference in degree centralities of the nodes in the two networks.
S, CC, T	Network density, number of connected components and number of triads respectively. For these measures, we calculate the difference between the gold and the predicted network. For example, a value of 6 in the column labeled CC and the row labeled Alice in Table 45 is the difference in number of connected components found in the predicted network and the gold network.

Table 47: Results comparing the two baseline systems (B-Simple and B-BOW) with the models trained on the tree kernel structures discussed in Section 14.3.2. $\uparrow$ means greater value is better. $\downarrow$ means lesser value is better. Network density of the gold network is 0.0166. The number of connected components in the gold network are 34. The number of triads in the gold network are 103.

	Stats					Centrality		Community		
System	P $\uparrow$	R $\uparrow$	F $\uparrow$	%A $\uparrow$	p $\uparrow$	D $\downarrow$	B $\downarrow$	S $\downarrow$	CC $\downarrow$	T $\downarrow$
B-Simple	0.40	1.00	0.57	97.58	0.0000	19.67	0.57	0.0245	23	439
B-BOW	0.47	0.87	0.61	98.15	0.0000	13.00	0.34	0.0145	14	165
PET	0.77	0.65	0.71	99.12	0.12	6.93	0.15	0.0026	2	0
GRW	0.61	0.68	0.65	98.78	0.38	8.31	0.10	0.0019	1	49
SqGRW	0.64	0.81	0.71	98.93	0.01	8.77	0.11	0.0045	6	34
PET.GR	0.72	0.60	0.66	98.96	0.15	7.75	0.12	0.0026	2	28
PET.GR _SqGRW	0.72	0.65	0.68	99.01	0.41	7.87	0.10	0.0016	3	18
GR.SqGRW	0.67	0.70	0.68	98.93	0.75	8.77	0.06	0.0008	1	23
GR.GRW _SqGRW	0.63	0.68	0.66	98.83	0.55	8.31	0.10	0.0013	0	6

1. **B-Simple**: For this baseline, we create a network by linking all pairs of entity mentions (of different entities) that appear in the same sentence.
2. **B-BOW**: This is the network extracted by building a model that uses bag-of-words features for training.

Discussion of results: Table 47 shows results for the SNA metrics (Section 14.3.3) for the kernel combinations used to extract a network from *Alice in Wonderland*. The terminology used in Table 47 is explained in Table 46.

Table 47 shows that all the systems trained using tree kernels are better than the two baselines across all SNA metrics. In terms of F-measure, both the baselines perform significantly worse than the tree kernel

based systems. In terms of p-value, both the baselines are significantly different from the gold network, whereas none of the tree kernel based systems are significantly different from the gold network. In terms of the distance between the vectors of centrality measures (D, B) for the predicted and gold network – the distance is larger for the baselines, which means that the difference in centrality measures of nodes in the baseline system and the gold network is larger than the difference in centrality measures of nodes in the other systems and the gold network. The difference in network densities of the baseline and gold network is also larger than the difference in network densities (S) of the other systems and the gold network. The same is the case with the number of connected components (CC) and the number of triads (T). Using these results, we conclude that the network predicted by our system that uses tree kernels performs well in terms of extracting an unweighted, undirected network from *Alice in Wonderland*. In particular, the tree structure derived from the phrase structure tree (PET) performs the best on most of the SNA metrics.

14.5 Conclusions

In this chapter, we have presented results for extracting a social network from *Alice in Wonderland* that is not restricted to interactions signaled by quoted speech. We have defined a social network for a fictional text as follows: nodes are characters and links are *social events*. Two nodes in the network are connected if the characters engage in a social event. We introduced the notion of *social events* in our previous work (Agarwal et al., 2010), in which we presented our annotation scheme for annotating social events on the Automatic Content Extraction (ACE-2005) corpus. We presented a preliminary system for social event detection and classification in Agarwal and Rambow (2010a). This system was trained and tested only on the ACE-2005 corpus. *A priori*, it is unclear if a system trained on a news corpus will be able to extract a high quality social network from a text from a very different genre (literary fiction). There are many syntactic and lexical differences between these genres. For example, news corpora have almost no questions, very little dialog presented as direct speech, and very little use of the first and second person pronouns. The vocabulary in literature can also be very different (relating to, say, whaling, passion, or teenage angst rather than current events). In this chapter, we have made two novel contributions. First, in an intrinsic evaluation, we have shown that our system without any domain (or genre) adaptation performs reasonably well on a new genre. Second, in an extrinsic evaluation, we have shown that the social network that our system extracts is not statistically distinguishable from the underlying gold network in terms of various standard and popularly used network analysis metrics. We conclude that our method is useful across genres.

Chapter 15

Argument Mining: Macro-level and Micro-level

15.1 Summary

Current studies in argument mining have focused primarily on the automatic identification of the structure of arguments (e.g., argument components and how these components are related to each other). These models of argumentation are referred to as monological or micro-level models, focusing on argument as *product*. In contrast, macro-level models (or dialogical models) and rhetorical models focus on the *process* of argument in a dialogue, being more suitable for analyzing negotiation or persuasion dialogues. We proposed a theoretically-grounded, response-centered approach for the discovery and classification of arguments and their (dis)agreement relations that has the potential to bring together macro-level and micro-level models of argumentation suitable for argument mining (Ghosh et al., 2014; Musi et al., 2017). Our approach relies on a two-level ontology that we argue could be the basis of a reliable annotation scheme, which in turn could be used for computational models. The first level (Upper Ontology) defines argumentative components as perceived by participants making reference to the intuitive notions of *CallOuts* and *Targets* and *Anaphoric Relationship* of the Agreement, Disagreement or Partial agreement and Partial disagreement types. The advantage is that of providing a framework to enable the discovery of unconventional structures and patterns of argumentation. We have shown that this upper ontology can be coded reliably by annotators (Ghosh et al., 2014). The second level (Lower Ontology) provides a classification of the type of “entities” which possibly constitute the targets of the (dis)agreement relations, drawing from traditional linguistic taxonomies of entities. Besides informing as to the sources of (dis)agreement, this lower level ontology intersects current NLP methodologies analyzing entities, events and relations. Under this framework we have conducted several studies. First, we looked at whether argumentation features derived from a coarse-grained argumentative discourse structure (claims, premises, support/attack relations) are good predictors of human-assigned scores of persuasive essays (Ghosh et al., 2016). Second, we extended this study to look at the correlation between the semantic type of claims (evaluation, interpretation, agreement/disagreement) and premises (logos, pathos, ethos) and persuasiveness of the message (Hidey et al., 2017). Third, we investigated whether the identification of argument schemes, namely the implicit and explicit inferential relations within and across argumentative units, is a feasible task that can tell us something about the relation between discourse units and argumentative units (Musi et al., 2016). We present briefly these three lines of inquiry. Findings from the work on argumentation have been contributing to the work on Social Influence.

Table 48: Argumentation Features

Feature Group	Id	Argumentation Feature Description
AC	1	# of Claims
	2	# of Premises
	3,4	# and fraction of sentences containing argument components
AR	5, 6	# and % of supported Claims
	7, 8	# and % of dangling Claims
	9	# of Claims supporting Major Claim
	10, 11	# of total Attacks and Attacks against Major Claim
TS	12	# of Argument <i>Chains</i>
	13	# of Argument <i>T ree</i> _{$h=1$}
	14	# of Argument <i>T ree</i> _{$h>1$}

15.2 Argumentation Features and Persuasive Essay Quality

15.2.1 Introduction

The goal of this study (Ghosh et al., 2016) was to investigate whether argumentation features derived from a coarse-grained argumentative structure of essays can help predict essay scores. We introduced a set of argumentation features related to argument components (e.g., the number of claims and premises), argument relations (e.g., the number of supported claims) and typology of argumentative structure (chains, trees). We showed that these features are good predictors of human scores for TOEFL essays, both when the coarse-grained argumentative structure is manually annotated and automatically predicted.

15.2.2 Methods, Assumptions and Procedures

To carry the study we use the argumentative discourse structure proposed by (Stab and Gurevych, 2014a): *argument components* (major claims, claims, premises) and *argument relations* (support, attack). The advantage of a simple annotation scheme is two-fold: more reliable human annotation, and better performance for argumentation mining systems designed to automatically identify the argumentative structure (Stab and Gurevych, 2014b). We introduced a set of argumentation features related to three main dimensions of argumentative structure: 1) features related to *argument components* such as the number of claims in an essay, number of premises, fraction of sentences containing argument components; 2) features related to *argument relations* such as the number and percentage of supported and unsupported claims; and 3) features related to the *typology of argumentative structure* such as number of chains and trees. Table 48 gives the details of all our proposed features.

Data And Annotation. We use a set of 107 essays from the TOEFL11 corpus that was proposed for the first shared task of Native Language Identification. The unit of annotation is a clause. Our annotated dataset, *TOEFL Arg*, includes 107 major claims, 468 claims, 603 premises, and 641 number of sentences that do not contain any argument component.

Automatically Detecting Claims/Premises. To automatically generate the argumentation features (Table 48), we first need to identify the argumentative structures: argument components (major claim, claim, and premise) and relations (support/attack). For argument component identification, we categorize clauses to one of the four classes (major claim (*MC*), claim (*C*), premise (*P*), and *None*). For argument relation identification, given a pair of argument clauses *Arg1* and *Arg2* the classifier decides whether the pair holds a support (*S*) or non-support (*NS*) relation (binary classification). For each essay, we extract all possible combinations of *Arg1* and *Arg2* from each paragraph as training data (654 *S* and 2503 *NS* instances;

attack relations are few and included in $N S$). We do not consider relations that may span over multiple paragraphs to reduce the number of non-support instances. For both tasks we use *Lexical* features (e.g., unigrams, bigrams, trigrams, modal verbs, adverbs, word-pairs for relation identification), *Structural* features (e.g., number of tokens/punctuations in argument, as well as in the sentence containing the argument, argument position in essay, paragraph position (paragraph that contains the argument)), *Syntactic* features (e.g., production rules from parse trees, number of clauses in the argument), and *Indicators* (discourse markers selected from the three top-level Penn Discourse Tree Bank (PDTB) relation senses: Comparison, Contingency, and Expansion).

15.2.3 Results and Discussion

When using the proposed argument features derived from the *manual* annotation of argument components and relations, we have shown that using all the argumentation features predicts essay scores that are highly correlated with human scores (0.803 quadratic weighted kappa (QWK)¹). Our findings show the “fraction of sentences containing argument components”, “number of supported claims”, and “number of $T_{reeh=1}$ structures” and “number of $T_{reeh>1}$ structures” have the highest correlation with high scoring essays, while a high “number and % of dangling claims” is a strong feature for low scoring essays. To fully automate the process we had to show whether the above results hold if the argumentative structure was *automatically* identified. Based on the automatic identification of the argument components and relations described above, we generate the argumentation features. Since our goal was to compare with the manual annotation setup, we trained on the dataset of 90 essays introduced by (Stab and Gurevych, 2014b) and tested on our TOEFL dataset. Overall, we see that we still have good correlation with human scores (0.73 QWK). However, a detailed analysis shows that argumentative features (especially argument relation and typology features) derived from the automatically produced argumentative structure, perform worse compared to the scores based on manual annotations. This is due to the wrong prediction of argument components, specifically wrongly labeling claims as premises, since both argument relations features and the typology features rely on correctly identifying the *claims*. When we extracted the argumentation features using *gold-standard* argument components and *predicted* argument relations, the correlation of argument related and typology features improved to 0.576 and 0.504, respectively (from 0.46 and 0.311), and the correlation of all features reached 0.769.

15.2.4 Conclusions

We showed that argumentation features derived from a coarse-grained, argumentative structure of essays are helpful in predicting essays scores that have a high correlation with human scores. Our manual annotation study shows that features related to argument relations are particularly useful. Our experiments using current methods for the automatic identification of argumentative structure confirms that distinguishing between claims and premises is a particularly hard task. This led to lower performance in predicting the essays scores using automatically generated argumentation features, especially for features related to argument relations and typology of structure.

15.3 Analyzing the Semantic Types of Claims and Premises

15.3.1 Introduction

Argumentative text has been analyzed both theoretically and computationally in terms of argumentative structure that consists of argument components (e.g., claims, premises) and their argumentative relations (e.g., support, attack). Less emphasis has been placed on analyzing the semantic types of argument components. In our work (Hidey et al., 2017), we proposed a two-tiered annotation scheme to label claims and premises and their semantic types in an online persuasive forum, *Change My View*, with the long-term goal

¹QWC corrects for chance agreement and it takes into account the extent of the disagreement between labels.

of understanding what makes a message persuasive. We aim to answer three questions: 1) can humans reliably annotate the semantic types of argument components? 2) are types of premises/claims positioned in recurrent orders? and 3) are certain types of claims and/or premises more likely to appear in persuasive messages than in non-persuasive messages? This line of work will be integrated in the work we are continuing on persuasive influence described in earlier chapters.

15.3.2 Methods, Assumptions and Procedures

We proposed and validated a systematic procedure to identify conceptual aspects of persuasion, presenting a two-stage annotation process on a sample of 78 threads from the sub-reddit *Change My View* (part of the corpus used for persuasive influence work described in earlier chapters). *Change My View* constitutes a suitable environment for the study of persuasive argumentation: users award a *Delta point* to the users that managed to change their views, thus providing a naturally labeled dataset for persuasive arguments. In the first stage, expert annotators are asked to identify claims and premises among the propositions forming the post. In the second stage, using crowdsourcing (Amazon Mechanical Turk) claims and premises are annotated with their semantic types. For premises, the semantic types are based on the Aristotelian modes of persuasion: *ethos* (credibility and trustworthiness of the speaker), *pathos* (appeal to audience feelings), and *logos* (appeal to the rationality of the audience through logical reasoning) or combination of them. For claims, we have considered two proposition types among those in Freeman's taxonomy (Freeman, 2000) that can work as claims since their truth is assailable, namely *interpretations* and *evaluations (rational/emotional)*. We have furthermore distinguished propositions expressing *agreement* and *disagreement* because they present an anaphoric function inherent to the dialogic nature of the corpus.

15.3.3 Results and Discussion

We present results to answer our three questions mentioned in the Goal of Work.

Can humans reliably annotate the semantic types of argument components? Our findings show that claims, premises and premise types can be annotated with moderate agreement (Krippendorff's $\alpha > 0.63$), while claim types are more difficult for annotators to reliably label ($\alpha = 0.46$). The 78 discussion threads comprise 278 turns of dialogue consisting of 2615 propositions in 2148 total sentences. Of these sentences, 786 contain a claim and 1068 contain a premise. Overall at the sentence-level, 36.5% of sentences contain a claim and 49.7% contain a premise. 22% of sentences contain no annotations at all. In terms of claims,² 15.8% of sentences contain a rational evaluation, 8.7% contain an interpretation, and 7.3% contain an emotional evaluation, while only 2.5% contain agreement and 2.3% contain disagreement. For premises, 44% contain logos, 29% contain pathos, and only 3% contain ethos.

Are types of premises/claims positioned in recurrent orders? We first report the results of the sequential transitions at the proposition level between types of claims (agreement, disagreement, rational evaluation, emotional evaluation, and interpretation) and premises (pathos, ethos, and logos, and their respective combinations). If the previous proposition is not labeled as a claim or premise, we set the previous category to "None." If the sentence is the start of a post, we set the previous category to "BOP" (beginning of post). We also include transitions to the end of the post (EOP). Our findings show that considering only claims at the beginning of the post, rational evaluations (23%), agreements (5%), and interpretations (13%) are more likely to appear at the start than in general. On the other end, premises expressing *pathos* are less likely to appear at the end of the post (only 7% of the time), while less surprisingly, unannotated sentences (farewell messages, for example) are more likely to appear at the end (20% of the time). As far as sequences of modes of persuasion, arguments expressing *logos* or *pathos* are more likely to occur consecutively (for logos, 46% following logos and 48% following pathos and for pathos, 31% and 34% respectively) than in the overall

²We took the majority vote among Turkers to determine the types of claims and premises.

distribution (37% *logos* and 24% *pathos*). Finally, *logos* is more likely to follow a rational evaluation (49% of the time) when compared to the overall distribution of *logos* and the same is true for emotional evaluations and *pathos* (39%). Second, we report the counts for the type of premise given its associated claim type in the post. Our findings show that premises classified as *pathos* are in support of rational evaluations 34% of the time that *pathos* occurs, while *logos* supports rational evaluations 38% of the time ($p < 0.05$) and *ethos* 28% of the time. Similarly, there is a slight preference ($p < 0.05$) for *pathos* to support evaluation-emotional claims, with 20% of *pathos* arguments supporting that type, 17% of *logos* arguments and 17% of *ethos* supporting it, respectively. Finally, authors demonstrate a preference for *logos* when addressing the claims of an author in the previous post ($p < 0.01$).

Are certain types of claims and/or premises more likely to appear in persuasive messages than in non-persuasive messages? To investigate whether certain types of claims/premises correlate with persuasive/non-persuasive messages (research question 3), we conduct a preliminary analysis of the relationship between claims and premises in different contexts, in winning vs. non winning arguments. We recompute the transition matrix and conditional claim/premise matrix by splitting the dataset according to whether the responding poster received a delta point or not. We first find that for the unigram distribution rational evaluations are *less* likely to be found in winning arguments with 9% of propositions in positive and 14% in negative ($p < 0.01$). When we consider the joint distribution of premise combinations, we find that *pathos* and *logos* are more likely to occur together in successful threads, with 23% and 17% respectively ($p < 0.01$). For the transition distribution, compared to positive threads, negative threads show fewer agreements opening up the posts ($p < 0.05$). Agreeing with what was previously said by another speaker before expressing a possibly divergent opinion constitutes a traditional persuasive rhetorical strategy. Moreover, positive threads are slightly more likely to show consecutive arguments of the same type (*logos/logos*; *pathos/pathos*) ($p < 0.01$), suggesting the hypothesis that conceptual coherence plays a role as persuasive strategy. Examining premise/claim patterns qualitatively, it seems that positive threads generally feature more interpretations, especially based on arguments of the *logos* type, at the expense of the number of evaluations. This type of claim/premise pattern is likely to be perceived as less subjective.

15.3.4 Conclusions

In this study we propose an annotation scheme for the identification of persuasive conceptual features. Compared to previous pilot works in the same vein, we distinguish different semantic types of premises and of claims with the long term goal of investigating their persuasive role. We empirically validate the devised procedure through a two-tiered annotation project on a sample of 78 threads from the subreddit Change My View. While the annotation of argumentative components (claims, premises) was carried out by expert annotators, for the annotation of semantic types of premises and claims we relied on crowdsourcing. The annotation of premises and claims achieves moderate agreement, in line with state-of-the-art results. The same applies to the semantic types of premises, showing improvement with respect to previous attempts. The identification of the semantic types of claims appears to be more difficult due to the confusion between interpretations and rational evaluations. We plan to improve the guidelines to account for this difficulty. In order to understand the persuasive role of the semantic types of claims and premises under study, we observe the recurrent combinations of argumentative components, their preferred position in the post and their distribution in winning and non winning threads.

15.4 Argument Schemes

15.4.1 Introduction

The goal of this work (Musi et al., 2016) was to investigate whether identification of argument schemes, namely the implicit and explicit *inferential relations* within and across argument components, is a feasible

task. The long term goal of this is to study the correlation between the three models of persuasion discussed above and argument schemes.

15.4.2 Methods, Assumptions and Procedures

We proposed a study to annotate argument schemes for the argumentative discourse structure introduced by (Stab and Gurevych, 2014b). As a theoretical model of argument schemes we used the *Argumentum Model of Topics* (AMT) (Rigotti and Morasso, 2010; Rigotti, 2009). Unlike other theoretical approaches, it offers a *taxonomic hierarchy* of argument schemes based on criteria which are distinctive and mutually exclusive and which appeal to *semantic properties of the states of affairs* expressed by premises/claims, and not to the logical forms (deductive, inductive, abductive) of arguments, whose boundaries are still debated. We conducted a pilot annotation study using both minimally trained non-expert annotators as well as trained annotators, using 9 schemes found at the middle level of the hierarchy (e.g., INTRINSIC:DEFINITIONAL, INTRINSIC:CAUSAL, INTRINSIC:MEREOLOGICAL, EXTRINSIC:OPPOSITION, EXTRINSIC:ALTERNATIVES, EXTRINSIC: ANALOGY, EXTRINSIC: PRACTICAL EVALUATION, COMPLEX:AUTHORITY, NOARGUMENT). We used a corpus of 30 short persuasive essays already annotated with premises, claims and supports/against relations (Stab and Gurevych, 2014b).

15.4.3 Results and Discussion

Our initial results show a fair inter-annotator agreement (IAA) (Fleiss $\kappa=0.3$), constituting one of the highest obtained IAA for such a complex labeling task of argument schemes. Our detailed analysis into the reasons for annotator disagreement has brought to light a finding of interest: the high frequency of non-argumentative relation (NOARGUMENT) between premise/claims and claims/major claims, which has been nonetheless labeled as "support" relation in previous work. There were two main reasons for NOARGUMENT labeling. First, among the claims-major claims pairs frequently the propositional content of the claim rephrases that of the major claim, such as in the pair "[There should not be any restriction on artists' work]CLAIM. [The artist must be given freedom]"MAJOR CLAIM. In these cases, the presence of a "supports" relation is justified if redundancy is considered as a stylistic strategy for achieving consensus on a certain stance; however, the claim as a linguistic entity does not work as an argument.

Second, the clause annotated as premise happened to work as an argument only if joined with another clause. This happens because the annotation of premises and claims in the original dataset of (Stab and Gurevych, 2014b) was done at the clause level, and while there could have been a discourse relation between the premise and the claim (such as ELABORATION), there was no argumentative relation. Our finding, supported also by (Stede et al., 2016), shows that sometimes there is a mismatch between argumentative discourse units, which tend to encompass multiple clauses or even sentences and elementary discourse units. In addition, (Stede et al., 2016) conducted a study of correlating argumentative relations (e.g., support, rebut, undercut) with discourse relations from RST and SDRT theories in a corpus of micro-text. In our current work (after DEFT) we are conducting a study to annotate argument scheme on this corpus of micro-text and correlate argument schemes with rhetorical relations.

15.4.4 Conclusions

We presented a novel set of guidelines for the annotation of argument schemes based on the Argumentum Model of Topics. This framework is advantageous since it offers a hierarchical finite taxonomy of argument schemes based on linguistic criteria which are highly distinctive and applicable to every context. We have conducted a pilot annotation study of 30 short persuasive essays. From the qualitative analysis of the confusion matrixes it has emerged that: 1) linguistic indicators of argument schemes constitute useful clues for the annotators only if specific to one argument scheme, otherwise they can be a source of confusion; 2) the reconstruction of inferential rules is highly relevant to enhancing annotators' choices and 3) among Intrinsic:Causal argument schemes the subtype "Efficient cause" is the easiest to identify. We have improved the

guidelines according to these results and tested them on a reduced sample of 10 essays with 2 trained non-expert annotators and one expert annotator. The inter-annotator agreement has significantly improved (fair agreement). The confusion matrix suggests that the frequency of non argumentative or ambiguous relations is the main cause of disagreement.

Part IV

Speech Processing

Chapter 16

Speech Processing

16.1 Summary

The research on speech aimed to identify acoustic features of anomalies in speech that are indicative of atypical interactions. A key focus was on automatic disfluency detection and analysis of disfluency patterns associated with different communication contexts. In this effort, we annotated data on a variety of speech sources with a goal of advancing disfluency detection algorithms for robust performance on multiple domains. We also conducted preliminary analyses to demonstrate how disfluencies provide information about social interactions.

16.2 Introduction

Speech disfluencies – including filled pauses, repetitions, repairs and false starts – are frequent events in spontaneous speech. While the rate of disfluencies varies with the speaker and context, one study observed disfluencies once in every 20 words, affecting up to one third of utterances (Shriberg, 1994). Disfluencies are important to account for, both because of the information that they provide about the speaker and because of the challenge that the disrupted grammatical flow poses for natural language processing of spoken transcripts. In parsing, for example, explicit modeling of disfluencies in decoding improves parser accuracy even when the parser is trained on spontaneous speech transcripts (vs. text) (Charniak and Johnson, 2001; Johnson and Charniak, 2004), and clean-up of disfluencies before training the parser in combination with disfluency detection gives further gains (Kahn et al., 2005).

Using the terminology introduced in (Shriberg, 1994), speech disfluencies have the following basic pattern:

[reparandum + {interregnum }repair]

where the reparandum is the portion of the utterance that is corrected or abandoned, + indicates the interruption point (IP) which marks the end of reparandum, the (optional) interregnum could include filled pauses, edit phrases (e.g. “I mean”), or other filler words (e.g. “you know”), and the repair (also optional) is the portion of the utterance that corrects the reparandum. Most work on disfluency detection has aimed simply at identifying the reparandum. However, since disfluency types can be distinguished based on the differences between the repair and the reparandum, our work has aimed at also detecting the reparandum.

In order to develop algorithms that are robust across multiple domains and also to better understand inter- and intra-speaker variability in production of disfluencies, we chose to work with multiple corpora that reflect a range of social contexts, which necessitated some data annotation. Thus, this research had three main contributions: i) creation of new disfluency-annotated data sources, ii) development of new methods for disfluency detection, including both the reparandum and the repair, that provide robust performance across

domains, and iii) preliminary analyses illustrating cross-domain differences that reflect social interaction differences.

16.3 Methods, Assumptions and Procedures

Disfluency Annotation: We experimented with four different English corpora: Switchboard (conversational speech between strangers), Callhome (conversational speech between family members), SCOTUS (oral arguments from the Supreme Court of the United States), and Financial Crisis Inquiry Commission (FCIC) hearings (two hearings involving Blankfein testifying at different stages of the investigation). These different sources provide different levels of familiarity among participants and different stakes in the outcomes of the discussions. The Switchboard corpus had previously been annotated, but we wanted to use a simpler approach to representing repetition (stutter-like) disfluencies, eliminating the complex nesting into a flattened sequence for multiple repetitions. (It is shown in (Ostendorf and Hahn, 2013) that this improves cross-domain reparandum detection performance.) A small subset of Switchboard was reannotated with the flattened representation, and this subset was used to train a model to automatically reannotate the remainder of the corpus with all other disfluencies constrained not to change. (The reannotation results were very high precision – it is not a difficult task.) For the other corpora, we hand-annotated 15 Callhome conversations, 13 SCOTUS oral arguments, and sections of two FCIC hearings. Pilot annotation was done on a fourth corpus (ATAROS) of two party negotiations with different scenarios reflecting low vs. higher stakes problems.

Disfluency Detection: There have been several prior studies on automatic disfluency detection using various approaches. Some use lexical transcripts alone; others also use prosodic features. While prosodic features are useful for detecting interruption points, e.g., (Liu et al., 2006), methods using only lexical features do almost as well as those with both prosodic and lexical features and lexical features are the most challenging for cross-domain modeling, so our focus was on use of transcripts only. Some methods are integrated with parsing (Charniak and Johnson, 2001; Johnson and Charniak, 2004; Lease et al., 2006; Hale et al., 2006; Miller, 2009; Miller et al., 2009; Zwarts et al., 2010), but because parsers are more sensitive to domain differences, we chose not to use this approach. We build on the discriminative tagging approach first explored in (Liu et al., 2006) using conditional random fields (CRFs) and augmented with integer linear program constraints in (Georgila, 2009). We extended this work in several ways. First, we augmented the state space, incorporating explicit correction states to make it possible to detect repair regions. Then, we explored new features aiming at improved cross-domain performance, focusing on correction disfluencies, since these are the most difficult to detect. The new features were investigated together with multi-domain training. We also developed an approach for semi-supervised learning for transferring the models to scenarios where transcripts have not been carefully annotated, which is the form that is generally available for most transcribed corpora.

16.4 Results and Discussion

Disfluency Detection: Performance improvements were obtained for all corpora and all disfluency types by introducing new features, particularly using a new language model likelihood ratio of fluent vs. disfluent models and a distance-based pattern match feature. Reducing the number of part-of-speech classes helped for most scenarios. The performance for the different corpora using only Switchboard for training is given in table 49. (The performance for SCOTUS is slightly higher when in-domain data is added to the Switchboard corpus.) The biggest performance gains are for the correction detection, where improvements were 5-7% absolute (15-20% relative). For the multi-domain training, the only available annotated training data beyond Switchboard was for the SCOTUS corpus. The added data was only useful for the SCOTUS corpus as training data, but it benefited all sources when used in combination with the Switchboard corpus to identify common disfluency words.

We also developed an approach for semi-supervised learning for transferring the models to transcripts

Table 49: Disfluency type detection F-score in different corpora using only the Switchboard corpus in training.

	F-score			
	Reparandum			Correction
Test corpora	Repetition	Other	Either	Other
SWBD (test)	95.3	58.7	83.1	50.3
CallHome	80.0	43.3	60.7	38.1
SCOTUS	90.7	39.4	78.6	38.7
FCIC	81.3	39.1	62.0	36.0

that do not carefully annotate transcripts, which is the form that is generally available for most transcribed corpora. For such data, many of the disfluent words are not transcribed, so disfluencies are hard to detect. For example, reparandum detection of disfluencies drops from $F=.77$ to $F=.13$ for text that is not carefully transcribed, with even more dramatic loss for the repetitions. Combining semi-supervised learning with text processing on Switchboard to mimic the transcription differences led to dramatic improvements with $F=.62$. The ability to label this data gives us the opportunity of analyzing much more data to identify patterns of disfluency associated with high stress and power differentials.

Disfluency Analysis: We analyzed disfluency rates for fillers, repetitions, restarts and repair disfluencies in the Switchboard, Callhome, SCOTUS, FCIC and ATAROS corpora as a function of speaker and context, finding:

- There is no one type of disfluency that is predictive of the rates of other types for an individual, and individual variation of disfluency rates is better modeled with a log scale than a linear scale.
- Despite the fact that both Switchboard and Callhome corpora include informal telephone conversations, the disfluency type rates are very different, likely due to the fact that the conversants know each other well in Callhome. This led to development of a modified disfluency annotation guideline.
- Contrasting disfluency rates for Blankfein in the two FCIC hearings, the utterance-initial repetition disfluency rates are significantly higher in the later hearing, which is often associated with content planning. In SCOTUS, a similar effect is found comparing transcripts of Roberts as a lawyer vs. as a justice.
- In the ATAROS data, we find higher rates of disfluencies in the higher stakes discussions, consistent with our findings with the Financial Crisis Inquiry Commission (FCIC) hearings.
- In the SCOTUS corpus, the disfluency rate of lawyers tends to be correlated with the case disfluency rate, but this is not the case for justices, suggesting that there is an effect of relative power in disfluency entrainment.
- Using entity counts as a proxy for cognitive load in the SCOTUS cases, there was no correlation with disfluency rates of judges. It may be that this is simply too crude an approximation to cognitive load, and/or that other factors are more important in this domain.

16.5 Resulting Resources

Resources resulting from this effort include the annotated disfluency data, and open source software for disfluency detection.

16.6 Conclusions

In summary, this work led to substantial advances in multi-domain disfluency detection, and the introduction of a method for repair detection that provides information about disfluency types. Analyses of the rates of different types of disfluencies in different corpora suggest that disfluency rates for different types carry information about the social situation (familiarity, power differentials) and stakes of the discussion. Thus, the improved disfluency detection algorithms are useful both for cleaning speech transcripts to facilitate natural language processing on speech, but also for providing information about social context in conversational interactions.

Part V

**Technology, Resources, Deliverables,
Publications**

Chapter 17

New Technology

17.1 Deliverables

We summarize the state of our software deliverables to the ADEPT framework developed as part of DARPA DEFT at BBN with participation from other DEFT performers. Adept is a scalable, multi-layered, plug-and-play framework. Adept enables the rapid integration of natural language processing (NLP) algorithms.

ADEPT Algo	DRATS Task	Activity Description	Status	Sites
6.1	A2	Belief	Releasable	Columbia, GW
	A2	Hedging		Columbia
6.2	A2	Sarcasm	Releasable	Columbia
6.3	A5	Sentence Similarity	In Progress	GW
6.5	A2	Sentiment	Working on benchmark test	Columbia
6.6	A3	Social Network Extr.	Working on regression test	Columbia
6.8	A3	Social Influence	BBN accepted proposed API; delivery of system planned	Columbia
	A3	Social Power		Columbia
6.10	A3	Social Recognition	Delivered, pending resolution of python issue with BBN	UW
6.11	A5	Entity Linking	In Progress	GWU

Note that 6.4 and 6.9 have been dropped, and 6.7 (Spanish Sarcasm Detector) has been combined with 6.2.

17.2 ADEPT Algorithms

We now summarize our ADEPT algorithms and list their relation to the tasks.

ADEPT Algo	Description	DRATS Task	Sites	PIs
6.1	Modality-CB Tagger	A2	Columbia, GW	Diab, Hirschberg, Rambow
6.2	Sarcasm	A2	Columbia	Muresan
6.3	Sentence Similarity	A5	GW	Diab
6.5	Opinion	A2	Columbia	McKeown
6.6	Social Event Extraction	A3	Columbia	Rambow
6.7	Sarcasm Spanish	A2	Columbia	Muresan
6.8	Social Influence and Power	A3	Columbia	McKeown, Rambow
6.10	Community Endorsement	A3	UW	Ostendorf
6.11	NER and linking	A5	GW	Diab
6.12	Event detection	A5	GW	Diab

Note that ADEPT algorithms 6.4 and 6.9 have been withdrawn.

17.3 Knowledge Base Population and Languages

Finally, we summarize the nature of these efforts in terms of Knowledge Base population (KB) and languages. The checkmark (X) in the column “KB Population” indicates that knowledgebase population is implemented in an online knowledge base developed at BBN for the purpose of showcasing integration.

DRATS Task	Activity Description	ADEPT Algo	KB Population	Languages P2	Languages P3	PIs
A2	Belief	6.1	Direct X	En	Sp, Ch	Diab, Rambow
A2	Hedging		Through Belief X	En, Sp	Ch	Hirschberg
A2	Sentiment	6.5	Direct X	En, Sp	Ch	McKeown
A2	Sarcasm	6.2, 6.7	Through Belief/Sentiment	En, Sp	Ch	Muresan
A3	Social Influence	6.8	Direct	En	Ch, Sp	McKeown
A3	Social Power			En	Ch, Sp	Rambow
A3	Community Endorsement	6.10	Direct X	En, Sp	Ch	Ostendorf
A3	Social Network Extr.	6.6	Direct	En, Ch	Sp	Rambow
A5	Sentence Similarity	6.3	Through other algorithms	En	Ch, Sp	Diab
A5	NER and Linking	6.11	Direct	-	En, Sp, Ch	Diab
A5	Event Detection	6.12	Direct	-	En, Sp, Ch	Diab
A5	Argumentation Mining	6.8	Through 6.8 and 6.1	En	Sp, Ch	Muresan

Chapter 18

Publications

We list some of the publications derived from DEFT-funded research by area of research. This list is not exhaustive.

18.1 Belief and Hedging

(Prabhakaran et al., 2012a), (Prokofieva and Hirschberg, 2014), (Prabhakaran et al., 2015a), (Werner et al., 2015a), (Colomer et al., 2016b)

18.2 Sentiment

(Rosenthal and McKeown, 2015), (Rosenthal and McKeown, 2016), (Rosenthal and Mckeown, 2017), (Hidey and McKeown, 2018)

18.3 Social Network Extraction

(Agarwal et al., 2014b), (Agarwal et al., 2014a)

18.4 Power

(Prabhakaran and Rambow, 2013a), (Prabhakaran and Rambow, 2014), (Prabhakaran et al., 2014b), (Prabhakaran et al., 2014a)

18.5 Sarcasm

(Ghosh et al., 2015a), (Muresan et al., 2016), (Ghosh et al., 2017)

18.6 Community Endorsement

(Jaech et al., 2015), (Fang et al., 2016), (Tran and Ostendorf, 2016), (Cheng et al., 2017), (He et al., 2017), (Zayats and Ostendorf, 2018)

Chapter 19

Acronyms

Acronym	Meaning
AMT	Argumentum Model of Topics
BOP	Beginning of Post
BoW	Bag-of-Words
CEP	Community Endorsed Person
CRF	Conditional Random Field
EL	Entity Linking
EOP	End of Post
FCIC	Financial Crisis Inquiry Commission
IAA	Inter-Annotator Agreement
IAC	Internet Argument Corpus
KB	Knowledge Base
KNN	K-Nearest Neighbor
LCB	Linguistic Committed Belief
LDC	Linguistics Data Consortium
LSSD	Literal/Sarcastic Sense Disambiguation
LSTM	Long Short- Term Memory
LSTM	Long Short-Term Memory
MTurk	Amazon Mechanical Turk
MVME	Maximum-value matrix element
MVMEwe	Maximum-value matrix element with word embeddings
NDCG	Normalized Discounted Cumulative Gain
NEAR	Named Entity Alias Recognition
NEL	Named Entity Linking
NER	Named Entity Recognition
PDTB	Penn Discourse Tree Bank
POS	Part-of-Speech
QWK	quadratic weighted kappa
SCOTUS	Supreme Court of the United States
STS	Semantic Textual Similarity
SVM	Support Vector Machine
SVM	Support Vector Machines
WTMF	Weighted Textual Matrix Factorization

Bibliography

- Samir AbdelRahman, Mohamed Elarnaoty, Marwa Magdy, and Aly Fahmy. Integrated machine learning techniques for arabic named entity recognition. *IJCSI*, 7:27–36, 2010.
- Ahmed Abdul-Hamid and Kareem Darwish. Simplified feature set for arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, NEWS '10, pages 110–115, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-78-7. URL <http://dl.acm.org.proxygw.wrlc.org/citation.cfm?id=1870457.1870474>.
- Apoorv Agarwal and Owen Rambow. Automatic detection and classification of social events. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1024–1034, Cambridge, MA, October 2010a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D10-1100>.
- Apoorv Agarwal and Owen Rambow. Automatic detection and classification of social events. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1024–1034. Association for Computational Linguistics, 2010b.
- Apoorv Agarwal, Owen C. Rambow, and Rebecca J. Passonneau. Annotation scheme for social network extraction from text. In *Proceedings of the Fourth Linguistic Annotation Workshop*, 2010. URL <http://www.aclweb.org/anthology/W10-1803>.
- Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. Social network analysis of alice in wonderland. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 88–96, Montréal, Canada, June 2012a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W12-2513>.
- Apoorv Agarwal, Adinoyi Omuya, Aaron Harnly, and Owen Rambow. A Comprehensive Gold Standard for the Enron Organizational Hierarchy. In *Proceedings of the 50th Annual Meeting of the ACL (Short Papers)*, pages 161–165, Jeju Island, Korea, July 2012b. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P12-2032>.
- Apoorv Agarwal, Sriramkumar Balasubramanian, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. Frame semantic tree kernels for social network extraction from text. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 211–219, Gothenburg, Sweden, April 2014a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/E14-1023>.
- Apoorv Agarwal, Daniel Bauer, and Owen Rambow. Using frame semantics in natural language processing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 30–33, Baltimore, MD, USA, June 2014b. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W14-3008>.
- Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibliographic search. *Commun. ACM*, 18(6):333–340, June 1975. ISSN 0001-0782. doi: 10.1145/360825.360855. URL <http://doi.acm.org/10.1145/360825.360855>.

- Rami Al-Rfou'. Spanish wikipedia dump. <https://sites.google.com/site/rmyeid/projects/polyglot>, 2013.
- Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*, 2013.
- Mohammed Attia, Ayah Zirikly, and Mona Diab. The power of language music: Arabic lemmatization through patterns. *COLING 2016*, page 40, 2016.
- Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics, 1998.
- A. Bavelas. A mathematical model for group structures. *Human Organization*, 7:16–30, 1948.
- Yassine Benajiba and Paolo Rosso. Arabic named entity recognition using conditional random fields. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC*, volume 8, pages 143–153. Citeseer, 2008.
- Yassine Benajiba, Paolo Rosso, and José-Miguel Benedí. Anersys: An arabic named entity recognition system based on maximum entropy. In *CICLing*, pages 143–153, 2007.
- Yassine Benajiba, Mona Diab, and Paolo Rosso. Arabic named entity recognition using optimized feature sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 284–293. Association for Computational Linguistics, 2008.
- Yassine Benajiba, Mona Diab, and Paolo Rosso. Arabic named entity recognition: A feature-driven study. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(5):926–934, 2009.
- Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.
- Shane Bergsma and Grzegorz Kondrak. Alignment-based discriminative string similarity. In *Annual meeting-Association for Computational Linguistics*, volume 45, page 656, 2007.
- Or Biran, Sara Rosenthal, Jacob Andreas, Kathleen McKeown, and Owen Rambow. Detecting influencers in written online conversations. In *Proceedings of the Second Workshop on Language in Social Media*, pages 37–45, Montréal, Canada, June 2012. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W12-2105>.
- Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003a.
- D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of machine Learning research*, 3:993–1022, 2003b.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
- Stephen P. Borgatti and Rob Cross. A relational view of information seeking and learning in social networks. *Management science*, 2003.
- Philip Bramsen, Martha Escobar-Molano, Ami Patel, and Rafael Alonso. Extracting social power relationships from natural language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 773–782, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P11-1078>.

- U. Brandes, J. Raab, and D. Wagner. Exploratory network visualization: Simultaneous display of actor status and connections. *Journal of Social Structure*, 2001.
- Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.
- Cristian Cardellino. Spanish Billion Words Corpus and Embeddings, March 2016. URL <http://crscardellino.me/SBWCE/>.
- R Caruna. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 41–48, 1993.
- E. Charniak and M. Johnson. Edit detection and parsing for transcribed speech. In *Proc. NAACL*, pages 118–126, 2001.
- Ying Chen and James Martin. Towards robust unsupervised personal name disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 190–198, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D/D07/D07-1020>.
- Hao Cheng, Hao Fang, and Mari Ostendorf. A factored neural network model for characterizing online discussions in vector space. In *Proc. EMNLP*, 2017.
- Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.
- Juan Pablo Colomer, Keyu Lai, and Owen Rambow. Detecting level of belief in chinese and spanish. In *Proceedings of the ExProM Workshop at Coling*, Osaka, 2016a. To appear.
- Juan Pablo Colomer, Keyu Lai, and Owen Rambow. Detecting level of belief in chinese and spanish. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics (ExProM)*, pages 22–30, Osaka, Japan, December 2016b. The COLING 2016 Organizing Committee. URL <http://aclweb.org/anthology/W16-5003>.
- Rob Cross, Andrew Parker, and Laurence Prusak. Knowing what we know:-supporting knowledge creation and sharing in social networks. *Organizational Dynamics*, 2001.
- Aron Culotta, Ron Bekkerman, and Andrew McCallum. Extracting social networks and contact information from email and the web. In *Proceedings of First Conference on Email and Anti-Spam (CEAS-1)*, 2004.
- Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. Echoes of power: language effects and power differences in social interaction. In *Proceedings of the 21st international conference on World Wide Web, WWW '12*, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187931. URL <http://doi.acm.org/10.1145/2187836.2187931>.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-1025>.
- Kareem Darwish. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1558–1567, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-1153>.

- Kareem Darwish and Wei Gao. Simple effective microblog named entity recognition: Arabic as an example. In *LREC*, pages 2513–2517, 2014.
- Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P11-1061>.
- Marlies Santos Deas, Or Biran, Kathleen McKeown, and Sara Rosenthal. Spanish twitter messages polarized through the lens of an english system. In *Proceedings of TASS workshop on Sentiment Analysis at SEPLN*, 2015.
- Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2017.
- Mona T Diab, Lori Levin, Teruko Mitamura, Owen Rambow, Vinodkumar Prabhakaran, and Weiwei Guo. Committed belief annotation and tagging. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 68–73. Association for Computational Linguistics, 2009.
- David K. Elson, Nicholas Dames, and Kathleen R. McKeown. Extracting social networks from literary fiction. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, 2010.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.
- Hao Fang, Hao Cheng, and Mari Ostendorf. Learning latent local conversation modes for predicting community endorsement in online discussions. In *Proc. International Workshop on Natural Language Processing for Social Media*, 2016.
- Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Muñoz. Applying wikipedia’s multilingual knowledge to cross-lingual question answering. In *Proceedings of the 12th International Conference on Applications of Natural Language to Information Systems, NLDB’07*, pages 352–361, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-73350-7, 978-3-540-73350-8. URL <http://dl.acm.org/citation.cfm?id=2394705.2394745>.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL ’05*, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219885. URL <http://dx.doi.org/10.3115/1219840.1219885>.
- Matthew Francis-Landau, Greg Durrett, and Dan Klein. Capturing semantic similarity for entity linking with convolutional neural networks. *arXiv preprint arXiv:1604.00734*, 2016.
- James B Freeman. What types of statements are there? *Argumentation*, 14(2):135–157, 2000.
- Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1 (3):215–239, 1979.
- Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*, 2017.
- K. Georgila. Using integer linear programming for detecting speech disfluencies. In *Proc. NAACL HLT*, 2009.

- Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 39–48, 2014.
- Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1003–1012, Lisbon, Portugal, September 2015a. Association for Computational Linguistics.
- Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1003–1012, Lisbon, Portugal, September 2015b. Association for Computational Linguistics. URL <http://aclweb.org/anthology/D15-1116>.
- Debanjan Ghosh, Aquila Khanam, Yubo Han, and Smaranda Muresan. Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 549–554, 2016.
- Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. The role of conversation context for sarcasm detection in online interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 186–196. Association for Computational Linguistics, 2017.
- Eric Gilbert. Phrases that signal workplace hierarchy. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, CSCW '12*, pages 1037–1046, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1086-4. doi: 10.1145/2145204.2145359. URL <http://doi.acm.org/10.1145/2145204.2145359>.
- Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*, pages 581–586. Association for Computational Linguistics, 2011.
- Stephen Guo, Ming-Wei Chang, and Emre Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N13-1122>.
- Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of 43th Annual Meeting of the Association for Computational Linguistics*, 2005.
- Nizar Habash and Owen Rambow. Magead: a morphological analyzer and generator for the arabic dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 681–688. Association for Computational Linguistics, 2006.
- Nizar Habash, Mona T Diab, and Owen Rambow. Conventional orthography for dialectal arabic. In *LREC*, pages 711–718, 2012.
- Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. Morphological analysis and disambiguation for dialectal arabic. In *Hlt-Naacl*, pages 426–432, 2013.
- P. Hage and F. Harary. *Structural models in anthropology*. Cambridge University Press, 1983.
- John Hale, Izhak Shafran, Lisa Yung, Bonnie Dorr, Mary Harper, Anna Krasnyanskaya, Matthew Lease, Yang Liu, Brian Roark, Matthew Snover, and Robin Stewart. PCFGs with syntactic and prosodic indicators of speech repairs. In *Proc. Coling-ACL*, 2006.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN

- 1931-0145. doi: 10.1145/1656274.1656278. URL <http://doi.acm.org/10.1145/1656274.1656278>.
- John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.
- Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. Extracting signed social networks from text. In *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing*, TextGraphs-7 '12, pages 6–14, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=2392954.2392956>.
- Hua He, Denilson Barbosa, and Grzegorz Kondrak. Identification of speakers in novels. *The 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, 2013.
- Ji He, Mari Ostendorf, and Xiaodong He. Reinforcement learning with external knowledge and two-stage q-functions for predicting popular reddit threads. *CoRR*, abs/1704.06217, 2017. URL <http://arxiv.org/abs/1704.06217>.
- Christopher Hidey and Kathleen McKeown. Persuasive influence detection: The role of argument sequencing. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, February 2-7, 2018, New Orleans, Louisiana, USA.*, 2018.
- Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, September 2017.
- Daniel S Hirschberg. Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)*, 24(4):664–675, 1977.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997a. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997b.
- Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011.
- Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.
- Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.
- Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1, COLING '02*, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1072228.1072282. URL <http://dx.doi.org/10.3115/1072228.1072282>.
- Aaron Jaech, Vicky Zayats, Hao Fang, Mari Ostendorf, and Hannaneh Hajishirzi. Talking to the crowd: What do people react to in online discussions? In *Proc. EMNLP*, 2015.
- Heng Ji, HT Dang, J Nothman, and B Hachey. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*, 2014.

- Heng Ji, Joel Nothman, and Ben Hachey. Overview of tac-kbp2015 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2015)*, 2015.
- Hongyan Jing, Nanda Kambhatla, and Salim Roukos. Extracting social networks and biographical facts from conversational speech transcripts. *45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic.*, 2007.
- Thorsten Joachims. Making Large-Scale SVM Learning Practical. In Bernhard Schölkopf, Christopher J.C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, USA, 1999. MIT Press.
- J. C. Johnson. Anthropological contributions to the study of social networks: A review. *Advances in social network analysis: Research in the social and behavioral sciences*, 1994.
- M. Johnson and E. Charniak. A tag-based noisy channel model of speech repairs. In *Proc. ACL*, 2004.
- J. G. Kahn, M. Lease, E. Charniak, M. Johnson, and M. Ostendorf. Effective use of prosody in parsing conversational speech. In *Proc. EMNLP-HLT*, pages 233–240, 2005.
- George Karypis. Cluto-a clustering toolkit. Technical report, DTIC Document, 2002.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, 2003.
- Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.
- D. Knoke. *Political Networks: The structural perspective*. Cambridge University Press, 1990.
- Laura M. Koehly and Victoria A. Shivy. Social network analysis: A new methodology for counseling research. *Journal of Counseling Psychology*, 1998.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL <http://dl.acm.org/citation.cfm?id=645530.655813>.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016a.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016b.
- Quo Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. International Conference on Machine Learning*, pages 3104–3112, 2014.
- Matthew Lease, Mark Johnson, and Eugene Charniak. Recognizing disfluencies in conversational speech. *IEEE Trans. Audio, Speech, and Language Processing*, 14(5):169–177, 2006.
- Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems*, 2010.
- Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.
- Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Trans. Audio, Speech and Language Processing*, 14:1526–1540, 2006.

- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014a. URL <http://www.aclweb.org/anthology/P/P14/P14-5010>.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014b. URL <http://www.aclweb.org/anthology/P/P14/P14-5010>.
- Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119206. URL <http://dx.doi.org/10.3115/1119176.1119206>.
- Paul McNamee, James Mayfield, Dawn Lawrie, Douglas W Oard, and David S Doermann. Cross-language entity linking. In *IJCNLP*, pages 255–263, 2011.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b.
- Tim Miller. Improved syntactic models for parsing speech with repairs. In *Proc. NAACL HLT*, 2009.
- Tim Miller, Luan Nguyen, and William Schuler. Parsing speech repair without specialized grammar symbols. In *Proc. ACL-IJCNLP*, pages 277–280, 2009.
- Saif M Mohammad and Peter D Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics, 2010.
- Franco Moretti. *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso, 2005.
- Junichiro Mori, Takumi Tsujishita, Yutaka Matsuo, and Mitsuru Ishizuka. Extracting relations in social networks from the web using similarity between collective contexts. In *The Semantic Web-ISWC 2006*, pages 487–500. Springer, 2006.
- Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, 2006a.
- Alessandro Moschitti. Making tree kernels practical for natural language learning. In *Proceedings of European chapter of Association for Computational Linguistics*, 2006b.
- Smaranda Muresan, Roberto Gonzalez-Ibanez, Debanjan Ghosh, and Nina Wacholder. Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology*, 2016. ISSN2330-1643.
- Elena Musi, Debanjan Ghosh, and Smaranda Muresan. Towards feasible guidelines for the annotation of argument schemes. In *Proceedings of the Third Workshop on Argumentation Mining. In Proceedings of ACL, 2016*, 2016.
- Elena Musi, Debanjan Ghosh, Mark Aakhus, Smaranda Muresan, and Nina Wacholder. Building an ontology of (dis)agreement space for argument mining. In *SIGDIAL/SEMDIAL 2017 Joint Session on Negotiation Dialog (position paper)*, 2017.

- David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.
- Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. Convolution kernels on constituent, dependency and sequential structures for relation extraction. *Conference on Empirical Methods in Natural Language Processing*, 2009.
- Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, Deborah A. Cai, Jennifer E. Midberry, and Yuanxin Wang. Modeling topic control to detect influence in conversations using nonparametric topic models. *Machine Learning*, pages 1–41, 2013.
- Philip V. Ogren, Philipp G. Wetzler, and Steven Bethard. ClearTK: A UIMA toolkit for statistical natural language processing. In *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP workshop at Language Resources and Evaluation Conference (LREC)*, 2008.
- Naoaki Okazaki. Crfsuite: A fast implementation of conditional random fields (crfs), 2007. URL <http://www.chokkan.org/software/crfsuite/>.
- Aidinoyi Omuya, Vinodkumar Prabhakaran, and Owen Rambow. Improving the quality of minority class identification in dialog act tagging. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 802–807, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N13-1099>.
- Shereen Oraby, Vrindavan Harrison, Ernesto Hernandez, Lena Reed, Ellen Riloff, and Marilyn Walker. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 2016 Conference on SIGDIAL*, 2016.
- Mari Ostendorf and Sangyun Hahn. A sequential repetition model for improved disfluency detection. In *Proc. Interspeech*, 2013.
- Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010. ISSN 1041-4347. doi: 10.1109/TKDE.2009.191. URL <http://dx.doi.org/10.1109/TKDE.2009.191>.
- Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholi, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M Roth. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2014.
- Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*, 2014.
- Ellie Pavlick and Joel Tetreault. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74, 2016.
- James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001, 2001.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.
- Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. Learning sentiment lexicons in spanish. In *LREC*, volume 12, page 73, 2012.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*, 2011.

- Vinodkumar Prabhakaran and Owen Rambow. Written Dialog and Social Power: Manifestations of Different Types of Power in Dialog Behavior. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 216–224, Nagoya, Japan, October 2013a. Asian Federation of Natural Language Processing. URL <http://www.aclweb.org/anthology/I13-1025>.
- Vinodkumar Prabhakaran and Owen Rambow. Written dialog and social power: Manifestations of different types of power in dialog behavior. In *Proceedings of the IJCNLP*, pages 216–224, Nagoya, Japan, October 2013b. Asian Federation of Natural Language Processing. URL <http://www.aclweb.org/anthology/I13-1025>.
- Vinodkumar Prabhakaran and Owen Rambow. Predicting Power Relations between Participants in Written Dialog from a Single Thread. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 339–344, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P14-2056>.
- Vinodkumar Prabhakaran, Owen Rambow, and Mona Diab. Automatic committed belief tagging. In *Coling 2010: Posters*, pages 1014–1022, Beijing, China, August 2010a. Coling 2010 Organizing Committee. URL <http://www.aclweb.org/anthology/C10-2117>.
- Vinodkumar Prabhakaran, Owen Rambow, and Mona Diab. Automatic committed belief tagging. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1014–1022. Association for Computational Linguistics, 2010b.
- Vinodkumar Prabhakaran, Michael Bloodgood, Mona Diab, Bonnie Dorr, Lori Levin, Christine D. Piatko, Owen Rambow, and Benjamin Van Durme. Statistical modality tagging from rule-based annotations and crowdsourcing. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*, pages 57–64, Jeju, Republic of Korea, July 2012a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W12-3807>.
- Vinodkumar Prabhakaran, Owen Rambow, and Mona Diab. Predicting Overt Display of Power in Written Dialogs. In *Human Language Technologies: The 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Montreal, Canada, June 2012b. Association for Computational Linguistics.
- Vinodkumar Prabhakaran, Ajita John, and Dorée D. Seligmann. Who had the upper hand? ranking participants of interactions based on their relative power. In *Proceedings of the IJCNLP*, pages 365–373, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. URL <http://www.aclweb.org/anthology/I13-1042>.
- Vinodkumar Prabhakaran, Ashima Arora, and Owen Rambow. Power of confidence: How poll scores impact topic dynamics in political debates. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, pages 77–82, Baltimore, Maryland, June 2014a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W14-2710>.
- Vinodkumar Prabhakaran, Emily E. Reid, and Owen Rambow. Gender and power: How gender and gender environment affect manifestations of power. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1965–1976, Doha, Qatar, October 2014b. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1211>.
- Vinodkumar Prabhakaran, Tomas By, Julia Hirschberg, Owen Rambow, Samira Shaikh, Tomek Strzalkowski, Jennifer Tracey, Michael Arrigo, Rupayan Basu, Micah Clark, Adam Dalton, Mona Diab, Louise Guthrie, Anna Prokofieva, Stephanie Strassel, Gregory Werner, Yorick Wilks, and Janyce Wiebe. A new dataset and evaluation for belief/factuality. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 82–91, Denver, Colorado, June 2015a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S15-1009>.

- Vinodkumar Prabhakaran, Tomas By, Julia Hirschberg, Owen Rambow, Samira Shaikh, Tomek Strzalkowski, Jennifer Tracey, Michael Arrigo, Rupayan Basu, Micah Clark, et al. A new dataset and evaluation for belief/factuality. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 82–91, 2015b.
- A. Prokofieva and J. Hirschberg. Hedging and speaker commitment. In *Proceedings of 5th International Workshop on Emotion, Social Signals, Sentiment & Linked Open Data*. LREC, 2014.
- Rakuten. Rakuten shopping. <http://www.rakuten.com/>, 2013a.
- Rakuten. Rakuten spanish. <http://www.rakuten.es/>, 2013b.
- Rakuten. Rakuten taiwanese. <http://www.rakuten.com.tw/>, 2013c.
- Owen Rambow, Tao Yu, Axinia Radeva, Sardar Hamidian, Alexander Fabbri, Debanjan Ghosh, Christopher Hidey, Tianrui Peng, Mona Diab, Kathleen McKeown, et al. The columbia-gwu system at the 2016 tac kbp best evaluation. In *Proceedings of the 2016 NIST TAC KBP Workshop*, 2016.
- Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL ’09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-29-9. URL <http://dl.acm.org/citation.cfm?id=1596374.1596399>.
- Eddo Rigotti. Whether and how classical topics can be revived within contemporary argumentation theory. In *Pondering on problems of argumentation*, pages 157–178. Springer, 2009.
- Eddo Rigotti and Sara Greco Morasso. Comparing the argumentum model of topics to other contemporary approaches to argument schemes: the procedural and material components. *Argumentation*, 24(4):489–512, 2010.
- Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’11, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL <http://dl.acm.org/citation.cfm?id=2145432.2145595>.
- Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.
- Sara Rosenthal and Kathleen McKeown. Detecting opinionated claims in online discussions. In *Proceedings of the 2012 IEEE Sixth International Conference on Semantic Computing*, ICSC ’12, pages 30–37, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4859-3. doi: 10.1109/ICSC.2012.59. URL <http://dx.doi.org/10.1109/ICSC.2012.59>.
- Sara Rosenthal and Kathleen McKeown. Columbia nlp: Sentiment detection of subjective phrases in social media. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Atlanta, Georgia, 2013.
- Sara Rosenthal and Kathleen McKeown. Detecting influencers in multiple online genres. *ACM Trans. Internet Technol.*, 17(2):12:1–12:22, March 2017. ISSN 1533-5399. URL <http://doi.acm.org/10.1145/3014164>.
- Sara Rosenthal and Kathy McKeown. I couldn’t agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 168–177, Prague, Czech Republic, September 2015. Association for Computational Linguistics. URL <http://aclweb.org/anthology/W15-4625>.
- Sara Rosenthal and Kathy McKeown. Social proof: The impact of author traits on influence detection. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 27–36, Austin,

- Texas, November 2016. Association for Computational Linguistics. URL <http://aclweb.org/anthology/W16-5604>.
- Sara Rosenthal, Apoorv Agarwal, and Kathleen McKeown. Columbia nlp: Sentiment detection of sentences and subjective phrases in social media. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*, Dublin, Ireland, August 2014.
- Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo. Automated social hierarchy detection through email network analysis. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 109–117, 2007a.
- Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J. Stolfo. Automated social hierarchy detection through email network analysis. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web Mining and Social Network Anal.* ACM, 2007b.
- Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- R. Sanjek. What is social network analysis, and what it is good for? *Reviews in Anthropology*, 1974.
- Kathleen McKeown Sara Rosenthal, Suvarna Bothe. Columbia nlp: Sentiment slot filling. In *Proceedings of the TAC KBP workshop in Gaithersburg*, Maryland, USA, 2014.
- Roser Saur¹ and James Pustejovsky. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 38(2):261–299, 2012.
- Helmut Schmid. Probabilistic part-of-speech tagging using decision trees, 1994.
- Nathan Schneider, Behrang Mohit, Kemal Oflazer, and Noah A Smith. Coarse lexical semantic annotation with supersenses: an arabic case study. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 253–258. Association for Computational Linguistics, 2012.
- S. B. Seidman. Structural consequences of individual position in nondyadic social networks. *Journal of Mathematical Psychology*, 1985.
- Khaled Shaalan. A survey of arabic named entity recognition and classification. *Comput. Linguist.*, 40(2): 469–510, June 2014. ISSN 0891-2017. doi: 10.1162/COLI_a_00178. URL http://dx.doi.org.proxygw.wrlc.org/10.1162/COLI_a_00178.
- Marvin E. Shaw. Group structure and the behavior of individuals in small groups. *The Journal of Psychology*, 38(1):139–149, 1954. doi: 10.1080/00223980.1954.9712925. URL <http://www.tandfonline.com/doi/abs/10.1080/00223980.1954.9712925>.
- Alfonso Shimbel. Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15(4):501–507, 1953. ISSN 0007-4985. doi: 10.1007/BF02476438. URL <http://dx.doi.org/10.1007/BF02476438>.
- E. Shriberg. *Preliminaries to a theory of speech disfluencies*. PhD thesis, Department of Psychology, University of California, Berkeley, CA, 1994.
- Georg Simmel. *The Sociology of Georg Simmel*. The Free Press, New York, 1950.
- Malcolm K. Sparrow. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social networks*, 1991.
- Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1501–1510, 2014a.

- Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*(Oct. 2014), Association for Computational Linguistics, p.(to appear), 2014b.
- Manfred Stede, Stergos Afantenos, Andreas Peldszus, Nicholas Asher, and Jérémie Perret. Parallel discourse annotations on a corpus of short texts. In *Proc. of LREC, Portoroz, 2016*, 2016.
- Tomek Strzalkowski, Samira Shaikh, Ting Liu, George Aaron Broadwell, Jenny Stromer-Galley, Sarah Taylor, Umit Boz, Veena Ravishankar, and Xiaoi Ren. Modeling leadership and influence in multi-party online discourse. In *Proceedings of COLING*, pages 2535–2552, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL <http://www.aclweb.org/anthology/C12-1155>.
- Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM, 2008.
- Paul Thompson and Christopher C. Dozier. Name searching and information retrieval. *CoRR*, cmp-lg/9706017, 1997. URL <http://arxiv.org/abs/cmp-lg/9706017>.
- Noel M. Tichy, Michael L. Tushman, and Charles Fombrun. Social network analysis for organizations. *Academy of Management Review*, pages 507–519, 1979.
- Kentaro Torisawa. A new perceptron algorithm for sequence labeling with non-local features. In *In Proceedings of EMNLP*, 2007.
- Trang Tran and Mari Ostendorf. Characterizing the language of online communities and its relation to community reception. In *Proc. EMNLP*, 2016.
- Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, 2010.
- Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.
- Mengqiu Wang and Christopher D Manning. Cross-lingual projected expectation regularization for weakly supervised learning. 2014.
- Gregory Werner, Vinodkumar Prabhakaran, Mona Diab, and Owen Rambow. Committed belief tagging on the factbank and lu corpora: A comparative study. In *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, pages 32–40, Denver, Colorado, June 2015a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W15-1304>.
- Gregory J Werner, Vinodkumar Prabhakaran, Mona Diab, and Owen Rambow. Committed belief tagging on the factbank and lu corpora: A comparative study. In *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, pages 32–40, 2015b.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210, 2005.
- Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *arXiv preprint arXiv:1705.08292*, 2017.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.
- Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016.

- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016.
- David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1072133.1072187. URL <http://dx.doi.org/10.3115/1072133.1072187>.
- Jen-Yuan Yeh and Aaron Harnly. Email thread reassembly using similarity matching. In *CEAS 2006 - The Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California, USA*, Mountain View, California, USA, July 2006.
- Tao Yu, Christopher Hidey, Owen Rambow, and Kathleen McKeown. Leveraging sparse and dense feature combinations for sentiment classification, 2017.
- Victoria Zayats and Mari Ostendorf. Conversation modeling on reddit using a graph-structured lstm. *Transactions of the Association for Computational Linguistics*, 6:121–132, 2018.
- Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.
- Ayah Zirikly and Mona Diab. Anear: Automatic named entity aliasing resolution. In *Natural Language Processing and Information Systems*, pages 213–224. Springer, 2013.
- Ayah Zirikly and Mona Diab. Named entity recognition system for dialectal arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 78–86, Doha, Qatar, October 2014a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W14-3610>.
- Ayah Zirikly and Mona Diab. Named entity recognition for dialectal arabic. *ANLP 2014*, page 78, 2014b.
- Ayah Zirikly and Mona Diab. Named entity recognition for arabic social media. In *Proceedings of naacl-hlt*, pages 176–185, 2015.
- Ayah Zirikly and Masato Hagiwara. Cross-lingual transfer of named entity recognizers without parallel corpora. In *Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics*, pages 390–396, 2015.
- Simon Zwarts, Mark Johnson, and Robert Dale. Detecting speech repairs incrementally using a noisy channel approach. In *Proc. Coling*, pages 1371–1378, 2010.