AN EMPIRICAL STUDY OF VERBAL IRONY; IDENTIFICATION, INTERPRETATION, AND ITS ROLE IN TURN-TAKING DISCOURSE

By

DEBANJAN GHOSH

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Communication, Information and Library Studies

written under the direction of

Smaranda Muresan, Ph.D.

and approved by

____

____

____

____

____

New Brunswick, New Jersey

May, 2018

ABSTRACT OF THE DISSERTATION

An Empirical Study of Verbal Irony; Identification, Interpretation, and its Role in Turn-taking Discourse

By DEBANJAN GHOSH

Dissertation Director:

Smaranda Muresan, Ph.D.

Human communication often involves the use of figurative language, such as verbal irony or sarcasm, where the speakers usually mean the opposite of what they say. In this dissertation, I address three problems regarding verbal irony: automatic identification of verbal irony and its characteristics from social media platforms, interpretation of verbal irony, and examining the role of verbal irony in identifying dis(agreement) relations in discussion forums. To automatically detect verbal irony I propose computational models that are based on theoretical underpinnings of irony. I first reframe the question of irony identification as a word-sense disambiguation problem to understand how particular target words are used in the literal or figurative sense. Next, I thoroughly analyze two characteristics of irony; irony markers, and irony factors. I propose empirical models to identify irony, irrespective of contextual knowledge as well as with conversation context. I also analyze the context to understand what triggers an ironic reply and perform user studies to explain the machine learning model predictions. Regarding the interpretation of irony, I offer a typology of linguistic strategies for verbal irony interpretation and link it to various theoretical linguistic frameworks. I design computational models to capture these strategies and present empirical studies aimed to answer two questions: (1) what is the distribution of linguistic strategies used by hearers to interpret ironic messages; (2) do hearers adopt similar strategies for interpreting the speaker's ironic intent? Finally, I turn to the application of irony to show how the use of irony-based features assists in identifying argumentative relations from discussion forums.

I perform this research on two types of social media datasets: self-labeled data (e.g., microblogging platforms such as Twitter and Reddit threads), and crowdsource-labeled corpus (e.g., Internet Argumentative Corpus).

Acknowledgments

This dissertation would not have been possible without the support of my advisor, Smaranda Muresan. Smara always gave me the freedom to pursue my research in the direction I wanted and was ready to share her ideas and recommendations. Even when I tended to deviate, she was patient and brainstormed with me on every occasion. Smara's extreme attention to detail in conducting experiments and writing is inspiring. It was a privilege to work together.

I am also thankful to my brilliant committee members. Mark Aakhus and Nina Wacholder introduced me to the fantastic world of argumentation research, and I will always cherish their comments, suggestions, and the exchange of ideas during SALTS group meetings. I am thankful to Nick Belkin for his critique, and ideas on how to sharpen the research questions. Thanks to Kathy McKeown for her suggestions and comments. More than a committee member, Kathy has always been an immense influence on me as an NLP researcher.

Apart from my committee member, I am indebted to Elisabeth Camp, who directed me to literature from linguistics as well as philosophy on figurative language. I am also grateful to the esteemed faculty of SCI, especially Marie Radford, Jen Theiss, and Marija Dalbello who taught me to think in interdisciplinary terms. I am thankful to Rohini Srihari who introduced me to the field of computational linguistics at the University of Buffalo while guiding my Masters thesis. Thanks to my wonderful colleagues at Thomson Reuters R&D group. In between my M.S. and the Ph.D. I spent a great time at Thomson Reuters working on many real-world NLP problems. Thanks to Francisco Pereira, my mentor at Siemens research who introduced me to the area of brain representation of semantic knowledge.

My cohorts at Rutgers University: Xiaofeng Li, Vanessa Kitzie, Robyn Kaplan,

Shahe Sanentz, Alexa Bolanos, Surabhi Sahay, Ralph Gigliotti have provided the intellectual environment and warm friendliness conducive to conduct research. I will always cherish the great time we had during our Ph.D. coursework. Likewise, I am incredibly lucky to be a part of the vibrant NLP research group at Columbia University and Center for Computational Learning Systems (CCLS). Thanks to Vinod Prabhakaran, Apoorv Agarwal, Or Biran, Heba Elfardy, Noura Farra, Daniel Bauer, Sara Rosenthal, and Chris Hidey for suggestions during presentations and NLP meetings. Extra thanks go to Elena Musi and Olivia Winn, fellow lab mates at Columbia University, for their numerous recommendations during various stages of this dissertation.

During my Ph.D., I collaborated with a brilliant group of undergraduate, graduate students and postdoctoral scholars. I thank Elena Musi for collaboration on multiple projects on argument mining and hope we will collaborate more in the future. I thank Weiwei Guo for pointing me to the word embedding approach during the word sense-disambiguation research, Anusha Balakrishnan for her brilliant work on detecting sarcasm across time-frame, Alex Fabrri for his diligent research on conversation context, Aquila Khanam and Yubo Han for their annotation and research on identifying arguments in persuasive writing, and Rituparna Mukherjee, Renato Vieira, Daniel Chaparro, and Pedro Perez Sanchez for their help in conducting various AMTurk experiments on cross-lingual sarcasm detection.

I am grateful to the DARPA DEFT grant as well as SALTS scholarship that supported my dissertation and numerous travels to conference venues. Thanks to Joan Chabrak, Allison Machiaverna, and Danielle Lopez who help graduate students at every stage of their Ph.D. at SCI. Likewise, many thanks to Idrija Ibrahimagic and Kathy Hickey at Columbia University who handled all the paperwork and conduct everything so smoothly.

In Calcutta, my parents Rama Ghosh and Nani Gopal Ghosh were surprised by my decision to leave the current job for pursue a Ph.D. but they always kept the faith in me even when I was uncertain. I am incredibly fortunate for their love and support. Thanks to my sisters Ratnabali Basu, Dipabali Acharjee, Chaitali Dutta, and Kakali Ghosh for everything! They were my first teachers in every aspect of my life. Thanks to my brothers-in-law, Kalyan Basu, Ashish Acharjee, Rajib Dutta, and Kaushik Ghosh for their supports. Big hug to my nieces, Ranjabati, Aishani, and Riddhi. Also a big thanks to my friends from the University at Buffalo: Indranil Sarkar, Radhika Barua, Souvik Chattopadhyay, Anindita Banerjee, Kaushik Roychoudhuri, Senjuti Gupta, Tarashankar Shaw, and Dinesh Kumar, for their love and support.

Finally, thanks to my partner, pillar, and best friend, Nabaparna Ghosh. Without her encouragement, frankly, I would not have started the Ph.D. journey. As a fellow researcher, she is an inspiration in so many ways. Besides that, her love and unconditional support have been the most vital in putting together this dissertation. And for all such reasons, this dissertation is dedicated to her.

Table of Contents

Abstract ii

Acknowledgments iv

List of Tables xi

List of Figures xiv

1. Introduction 1

1.1. Overview 1

1.1.1. Irony Identification 4

1.1.2. Irony Interpretation 6

1.1.3. Role of Irony to Detect Dis(agreement) Relations 7

1.2. Irony and its Characteristics 8

1.3. Contributions 10

1.4. Structure of the Chapters 13

2. Data 15

2.1. Overview 15

2.2. Datasets 17

2.2.1. Twitter Dataset 17

2.2.2. Reddit Corpus 20

2.2.3. Internet Argument Corpus 21

3. Verbal Irony Identification 25

3.1. Overview 25

3.2. Background and Related Work 26

3.2.1. Theoretical Approaches of Irony Analysis 26
3.2.2. Computational Approaches for Irony Identification 29
3.3. Introduction to Irony Markers and Irony Factors 32
3.3.1. Theoretical Overview of Irony Markers 33
3.3.1.1. Tropes as Irony Markers: 34
3.3.1.2. Morpho-syntactic Irony Markers: 34
3.3.1.3. Typographic Irony Markers: 35
3.3.2. Theoretical Overview of Irony Factors 35
3.4. Data and Preprocessing 37
3.4.1. Twitter Data 38
3.4.2. Internet Argument Corpus 38
3.4.3. Reddit Corpus 39
3.4.4. Dense Representation of Input Data 39
3.5. Utterance-level Analysis 41
3.5.1. Empirical Analysis of Irony Markers 41
3.5.1.1. Irony Markers as Discriminative Features 43
3.5.1.2. Generalization of Irony Markers 48
3.5.2. Modeling of Irony Factors 51
3.5.2.1. Literal/Ironic Sense Disambiguation (LISD) task . . . 51
3.5.2.1.1. Identifying Target Words: 52
3.5.2.1.2. Disambiguation Task: 54
3.5.2.2. Identifying Incongruence at the Utterance Level . . . 61
3.6. Role of Conversation Context in Verbal Irony Identification 67
3.6.1. Computational Models and Experimental Setup 71
3.6.2. Results and Discussion 76
3.6.2.1. Prior Turn as Conversation Context 76
3.6.2.2. Prior Turn and Subsequent Turn as Conversation Context 81
3.6.3. Qualitative Analysis 82

3.6.3.1. Crowdsourcing Experiments 83
3.6.3.1.1. Crowdsourcing Experiment 1 83
3.6.3.1.2. Crowdsourcing Experiment 2 84
3.6.3.2. Comparing Turkers' Answers with Attention Models 85
3.6.3.3. Interpretation of Turkers' Answers and Attention Models 86
3.6.3.3.1. Semantic Coherence between Prior Turn and Current Turn 87
3.6.3.3.2. Attention Weights and Irony Markers 88
3.6.3.3.3. Incongruity between Conversation Context (P_TURN) and Current Turn (C_TURN) . . . 89
3.7. Conclusion 91
4. Verbal Irony Interpretation 93
4.1. Overview 93
4.2. Related Work and Background 95
4.3. Datasets of Speaker's Ironic Messages and Hearer's Interpretations . . . 97
4.4. Interpreting Verbal Irony: A Typology of Strategies 99
4.4.1. Linguistic Strategies 100
4.4.2. Links to Theoretical Frameworks 102
4.5. Empirical Analysis of Interpretation Strategies 103
4.5.1. Computational Methods 103
4.5.2. Results and Distribution of Linguistic Strategies 110
4.6. Discussion: Hearer-dependent Interpretation Strategies 112
4.6.1. Graded Interpretation: 113
4.7. Discussion: Message-dependent Interpretation Strategies 114
4.7.1. Message Interpreted the Same by All Hearers: 115
4.7.1.1. Relation between Difficulty of the Interpretation Task and Linguistic Strategies 116
4.8. Conclusions 117

5. The Role of Irony in Detecting Agree/Disagree Relations 119

5.1. Overview 119

5.2. Data 121

5.3. Related Work 122

5.4. Experiments 124

5.4.1. Experiments with Arg_{feats} Features 125

5.4.2. Experiments with $Irony_{feats}$ Features 129

5.5. Conclusion 133

6. Conclusions and Future Work 134

6.1. Contributions 134

6.2. Limitations 137

6.3. Future Work 138

List of Tables

1.1. Use of irony markers in two social media platforms 5

1.2. Examples of ironic and non ironic utterance 5

1.3. Example of speaker's ironic messages (I$_{im}$) and interpretations given by 3 Turkers (H$^{i}_{int}$). 7

1.4. Ironic messages from userB and their respective prior turns from IAC. . 8

2.1. Examples of ironic and non ironic utterances 18

2.2. Sentiment Hashtags for Collecting Tweets 19

2.3. Tweet Datasets for different irony identification and interpretation research 19

2.4. Examples from $Reddit$. We show sarcastic posts (posts from userB) as well as their conversation context (posts from userA) 21

2.5. Examples from IAC corpus. We show ironic posts as well as their conversation context here. 23

2.6. Different corpus and associated experiments (u denotes utterances and c denotes context 24

3.1. Different corpus and associated experiments (u denotes only the utterances whereas $u + c$ denotes utterances with their context) for verbal irony identification 38

3.2. Ablation tests of irony markers (I vs. NI_{rand}). **bold** are best scores (in %). 45

3.3. Ablation tests of irony markers (I vs. NI_{sent}). **bold** are best scores (in %). 46

3.4. Ablation Tests of irony markers for $Reddit$ posts. **bold** are best scores (in %). 46

3.5. Irony markers based on feature weights for $Twitter$ 47

3.6. Irony markers based on feature weights for $Reddit$ 47

3.7. Frequency of irony markers in ironic utterances from $Twitter$ and $Reddit$ platforms. The mean and the Standard Deviation (SD, in bracket) are reported. 48

3.8. Frequency of irony markers in different genres (subreddits). For brevity, Morpho_Syntactic is represented as MS. The mean and the Standard Deviation (SD, in bracket) are reported.$^{x^{**}}$ and $^{x^{*}}$ depict significance at $p \leq 0.005$ and $p \leq 0.05$, respectively. 49

3.9. Examples of target words(underlined) and their senses 53

3.10. Target words and # of training instances per class 54

3.11. Target words and their context words 56

3.12. Evaluation of distributional approaches (PMI and word embedding) for LISD experiments 59

3.13. Evaluation of classification approaches (SVM_{bl} and $kernel_{we}$) for LISD experiments. $kernel_{we}$s are abbreviated as k_{we} for brevity. 60

3.14. Evaluation of classification approaches for sense-disambiguation experiments (highest scores are **bold**) 67

3.15. Ironic turns (C_TURN) and their respective prior turns (P_TURN) in Twitter and discussion forums. 67

3.16. Ironic messages (C_TURNs) and their respective prior turns (P_TURN) and subsequent turns (S_TURN) from IAC_{v2}. 70

3.17. Ironic messages (C_TURNs) and their respective prior turns (P_TURN) and subsequent turns (S_TURN) from $Reddit$. 70

3.18. Experimental results for the discussion forum dataset (IAC_{v2}) (**bold** are best scores) 77

3.19. Experimental results for Twitter dataset (**bold** are best scores) 79

3.20. Experimental results for $Reddit$ dataset (**bold** are best scores) 80

3.21. Experimental results for training on $Reddit$ dataset and testing on IAC_{v2} using the best LSTM models (sentence-level attention). 81

3.22. Experimental results for $IAC_{v2_{st}}$ dataset using post prior and succeeding turns as context (**bold** are best scores) 82

4.1. Examples of speaker's ironic messages (I$_{im}$) and interpretations given by 3 hearers/Turkers (H$^{i}_{int}$). 94

4.2. Typology of linguistic strategies and their distribution (in %) over the dev set 102

4.3. Negation Lexicons 106

4.4. Distribution of interpretation strategies on two datasets (numbers and %) 110

4.5. Evaluation of Computational Methods on dev, $test$ and $SIGN_{test}$ set (in %) 110

4.6. Stereotypical negative situation clusters 116

4.7. Task difficulty and linguistic strategies 117

5.1. Ironic messages (C_TURNS) and their respective prior turns (P_TURN) from IAC. 120

5.2. Dis(agreement) categories with irony/non-irony data (in %) 122

5.3. The effect of various Arg_{feats} for the dis(agreement) categories via ablation tests. P/R/F1 and macro-average (F1) are reported. Best numbers are in **bold**. 128

5.4. The effect of various $Irony_{feats}$ for the dis(agreement) categories via ablation tests. P/R/F1 and macro-average (F1) are reported. Best numbers are in **bold**. 131

List of Figures

3.1. Ironic utterance with emoji (best in color) 35

3.2. Examples of word embeddings via t-SNE ((a): “great”; (b) “:)” 41

3.3. Example of the CNN_T architecture (Figure is inspired by Kim (2014)) 64

3.4. Sentence-level Attention Network for prior turn pt and current turn ct. Figure is inspired by Yang et al. (2016) 74

3.5. Conditional LSTM Network for prior turn pt and current turn ct; Figure is inspired by the model proposed in Rocktäschel et al. (2015) 75

3.6. Crowdsourcing Experiment 1: (a) number of different trigger selections made by the five turkers (1 means all Turkers selected the exact same trigger(s)) and (b) distribution of the number of sentences chosen by the Turkers as triggers in a given post; both in % 84

3.7. Crowdsourcing Experiment 2: (a) number of different trigger selections made by the five turkers (1 means all Turkers selected the exact same trigger(s)) and (b) distribution of the number of sentences chosen by the Turkers as triggers in a given post; both in % 85

3.8. Sentences in P_TURN; heatmap of the *attention weights* (LHS) and *Turkers' selection* (RHS) of which of those sentences trigger the ironic C_TURN=“Well, it's not as though you hold your tongue all that often when it serves in support of an anti-gay argument.” 86

3.9. Sentences in P_TURN; heatmap of the *attention weights* (LHS) and *Turkers' selection* (RHS) of which of those sentences trigger the ironic C_TURN). 87

3.10. Sentences from P_TURN that trigger verbal irony (Top) and Sentences from C_TURN that express verbal irony (Bottom). Tables show respectively the text from P_TURN and C_TURN (top and bottom) and Figure shows the heatmap of *attention weights* (LHS) and *Turkers' selection* (RHS) 88

3.11. Sentence from P_TURN and the heatmaps of the *attention weights* (LHS) and *Turkers' selection* (RHS) emphasizing which of these sentences trigger the ironic C_TURN 89

3.12. Attention visualization of semantic coherence between c and r 89

3.13. Attention visualization of incongruity between P_TURN and C_TURN . . 90

3.14. Attention visualization of incongruity between P_TURN and C_TURN . . 90

3.15. Attention visualization of incongruity between P_TURN and C_TURN . . 91

3.16. Sentences from P_TURN that trigger verbal irony (Top) and Sentences from C_TURN that represents verbal irony (Bottom). Tables show respectively the text from P_TURN and C_TURN (top and bottom) and Figure shows *attention weights* (LHS) and *Turkers' selection* (RHS) . . 91

4.1. Strategies (in%) selected by top three Turkers who attempted 500 HITs (best in color) 112

4.2. Strategies selected per message (in %) 114

5.1. F1 scores of dis(agreement) for different settings with Arg_{feats} (better in colour) 129

5.2. F1 scores of dis(agreement) for different settings with $Irony_{feats}$ (better in colour) 132

Chapter 1
Introduction

1.1 Overview

Natural Language Understanding (NLU) is an area of Computational Linguistics that aims to give computers the ability to process and understand human language. Applications range from conversational agents, detecting people's attitudes such as their sentiments and beliefs, to web-based search and question answering. Equipping computers with the ability to process and understand language is a hard problem for many reasons. First, natural language is ambiguous, both at the lexical level and the syntactic level. Lexical ambiguity arises since the same word may have multiple meanings (i.e., polysemy). For instance, the word "bank" can mean a financial institution, but it could also mean "river bank" ("we stood on the river bank") or "blood bank". In this case, based on the usage/context of the word "bank" we need to identify its sense such as whether it has been used to depict a financial institution or the bank of a river. Likewise, syntactic ambiguity appears because a sentence can have multiple parse trees. For example, consider the following sentence, "I saw the man with a telescope". Here, the prepositional phrase "with a telescope" can attach to either of the two noun phrases, "I" or "the man". Second, a concept or an entity can be described using multiple verbalizations, such as, "HRC", "Hillary", "Ms. Clinton" of the same entity "Hillary Clinton". Third, meaning can be *implicitly* expressed in text. In this dissertation, we focus on this third challenge of Natural Language Understanding, that is the modality of implicit meaning.

An area of Natural Language Understanding that has seen a lot of progress has been the identification and extraction of the propositional aspects of meaning that *explicitly* stated in a text. Systems that study propositional aspects of meaning typically extract *factual* information, such as “who” (agent) did “what” (theme) to “whom” (target) at “where” (location). Semantic role labeling and semantic parsing tasks are used to extract such factual information. Consider the following examples:

(1) Mary went to the movie.

(2) John is reading a novel.

Here, a semantic role labeling system (i.e., shallow semantic parser) could identify that “Mary” and “John” are the agents of the respective events “going to the movie” and “reading a book”. Computational approaches derive the meaning of such utterances by applying the “principle of compositionality” (sometimes called “Frege’s Principle”) (Partee, 1984). The principle states that,

> The meaning of an expression is a function of the meanings of its parts and the way they are syntactically combined.

However, understanding language often involves detecting meaning that is *implicit* or *indirect* and different from the literal meaning. This type of implicit meaning is usually categorized under the umbrella term — “extra-propositional aspects of meaning” – and often featured in human communication as figurative language.[1] Figurative (or non-literal) utterances include additional information associated with the contextual situation in which they occur (Regel, 2009).[2]

Roberts and Kreuz (1994) have defined a total of eight distinct types of figurative use of language. They are, hyperbole (i.e., exaggeration), idiom, indirect request, understatement, metaphor, rhetorical question, simile, and irony.[3] Use of such figurative language is ubiquitous in natural language text (Shutova, 2011). In this dissertation, we focus on one particular type of figurative language, verbal irony. Irony is common to all forms of discourse, be it literary or scientific. It is used in novels, poems, periodicals, films, and in popular culture. Moreover, the use of irony is common on social-media

[1] Extra propositional aspects also cover areas such as detection of hedging, factuality or belief, to name a few.

[2] The expression “creative thought” is also used sometime to represent the figurative language genres (Shutova, 2011).

[3] Note, hyperboles, metaphors, and rhetorical questions are sometime used as indicators of irony. See the discussion on irony markers in Section 3.3.1.

platforms such as Twitter and discussion forums (e.g., Reddit). Verbal irony is a subtype of irony that takes shape when a speaker's intended meaning is the opposite of what they are saying. This dissertation applies *computational models* to study how verbal irony is expressed in text.

A Natural Language Understanding system that is not able to detect verbal irony would fail to properly understand people's attitudes such as sentiments and beliefs. Consider the following examples.

(3) I love spending Saturday night in the emergency room!!

(4) Shooting in Oakland? That NEVER happens.

In example (3), although the positive sentiment word "love" is used, the author is ironic, and the intended sentiment is negative. Likewise, in example (4), the author is ironic and believes that shootings frequently occur in Oakland.

Automatic processing of figurative language such as irony, however, is different from recognizing literal meaning in many aspects. First, the principle of compositionality cannot be used directly in the figurative language. In example (3), the use of positive sentiment such as "love" does not directly agree with the circumstance described in the utterance, that is, spending a weekend in the emergency room. In fact, the intended meaning of the utterance is the *opposite* of the literal meaning (i.e., disliking the situation of spending Saturday in the emergency room). For both examples, an off-the-shelf sentiment and belief detection system will also probably fail since they might not detect any irony in the utterance. Second, irony needs to be inferred using pragmatic interpretation. Pragmatic interpretation is the process by which language users systematically apply information about social convention and conversation context to infer aspects of speaker intention that are not reflected in literal sentence meaning. For (3), the social convention informs us that "visiting the emergency room during the weekend" is not something that we associate with "love". Likewise, pragmatic interpretation is applied to understand irony when people are engaged in a conversation. Consider the following example of a dialogue on Twitter:

(5) **UserA**: *plane window shades are open during take-off and landing so that*

> *people can see if there is fire in case of an accident*
>
> **UserB**: *@UserA awesome! one more reason to feel really great about flying.*

Here, UserB's utterance is ironic, but without the conversation context, irony cannot be detected.

Most people can recognize irony from the above utterances (3)-(5) without difficulty. For an automatic Natural Language Understanding system, however, to carry out such a task is challenging. To a minimum, the system needs to detect the *semantic incongruency* between "fire in case of an accident" (UserA's utterance) and "feel really great about flying" (UserB's utterance), which is a characteristic of irony.

In this dissertation, through a combination of theoretically grounded computational models and user studies we investigate (1) how to *identify* verbal irony and its characteristics in social media, (2) how do people *interpret* verbal irony, and (3) what is the *role of irony* in detecting agreement and disagreement between participants in a conversation. We discuss them in the following sections.

1.1.1 Irony Identification

In the last couple of years, verbal irony detection has received a lot of attention in computational linguistics (Davidov et al., 2010; Liebrecht et al., 2013; Maynard and Greenwood, 2014; Wallace et al., 2014; Zhang et al., 2016; Ghosh and Veale, 2016; Schifanella et al., 2016). Most of the current research treats the problem as a general text classification problem to identify whether an utterance is ironic or not. Moreover, these approaches are not based on theoretical frameworks of irony with the notable exception of Riloff et al. (2013); Joshi et al. (2015); Bamman and Smith (2015).

In this dissertation we investigate two aspects of verbal irony studied in linguistics: *irony markers* and *irony factors* (Attardo, 2000a; Burgers, 2010) and develop theoretically grounded computational models for irony identification. Irony markers are indicators of irony that alert the readers that an utterance is ironic. We compare their usage across different social media platforms and identify specific markers that are associated with a particular platform. Table 1.1 presents a couple of ironic utterances with

Platform	Utterances
Discussion Forum (*Reddit*)	Are you telling me my iPhone 5 is only marginally better than iPhone 4S? I thought we were reaching a golden age with this game-changing device.
Twitter	With 1 followers I must be AWESOME. :P
Twitter	Stepping on tic tacs feel sooooo good :(

Table 1.1: Use of irony markers in two social media platforms

markers (hyperbole, such as "game-changing device", the elongated word "sooooo", the uppercase word "AWESOME", tag question "are you", etc.).

In contrary, irony factors, point to the inherent characteristics of irony, such as semantic incongruence, that occurs when the ironic situation (e.g., "visiting the emergency room") is incongruent with the sentiment of the utterance (e.g., "love"). We study two irony factors in this dissertation: "reversal of valence" and "semantic incongruence". The first factor informs us that the ironic meaning is opposite of the intended meaning of irony. To analyze the first factor, we reframe the detection of irony to a word/multi-word sense disambiguation problem. Two challenges need to be addressed here. First, how to collect a set of target words that can have either literal or ironic meaning, depending on context, and second, given an utterance and a target word, how to identify whether the sense of the target word is literal or ironic. We first propose an unsupervised alignment technique to retrieve specific target words (e.g., "love", "brilliant", "shocked") that can have both an ironic and a literal sense depending on the context. Next, given an unknown utterance and a target word we identify whether the use of the target word is literal or ironic. Table 1.2 presents two examples. Here, the task is to determine that the target word *love* is used in an ironic sense in the top example and used in its literal sense in the below example. We compare several distributional semantics methods and show that using word embeddings in a modified SVM kernel achieves the best results (i.e., 7-10% improvement in accuracy over a robust lexical baseline).

To analyze the second irony factor, i.e., the semantic incongruence, we conduct two separate experiments. In the first, we employ a novel architecture of dual Convolutional

Sense	Utterance
Irony	love going to the hospital on birthday
Not Irony	all want for christmas is fell in love

Table 1.2: Examples of ironic and non ironic utterance

Neural Networks (CNN) (LeCun and Bengio, 1995) that focuses on modeling the composition between the target ("love") and context ("visiting the emergency room") to detect the incongruence. Second, we use conversation context (e.g., prior turn in an online discussion) for irony identification and show that computational models that consider the conversation context outperform the model that models only the ironic post. We discuss our research on irony markers and irony factors in detail in Chapter 3. We have used two social media platforms, Twitter and discussion forums (e.g., $Reddit$, internet argument corpus).

1.1.2 Irony Interpretation

Apart from the identification of various aspects of irony, in this dissertation, we also examine how humans *interpret* irony. Verbal irony is always intended for an audience, and thus, we argue that besides *recognizing the speaker's ironic intent* it is equally important to understand *how the hearer interprets* the ironic message. We conduct experiments on the Amazon Mechanical Turk (MTurk) platform where we ask the annotators to rephrase ironic messages to express the intended meaning of the message. The rephrasings are the verbalizations of how hearers interpret irony, and we utilize the rephrasings for our analysis. Table 1.3 shows an example where given an ironic message I_{im} (e.g., "It's encouraging how Portland Police Officers feel they're above the law") three Turkers rephrase that to represent the intended meanings (e.g., H^{i}_{int}). For the ironic utterance, the strength of negative sentiment perceived by the hearer depends on how they interpret the speaker's intended meaning. For instance, Turkers may prefer to use direct antonyms (e.g., "encouraging" $\rightarrow$ "discouraging") or negate the sentiment (e.g., "encouraging" $\rightarrow$ "not encouraging"); thus, the negative sentiment perceived in the first case might be higher than the latter. We propose a new typology of linguistic strategies to categorize the rephrasings posted by the Turkers. We empirically validate these strategies and investigate the distribution of the strategies in two corpora. We also look at when hearers adopt similar or dissimilar strategies to interpret the speaker's ironic intent.

Irony interpretation research is primarily correlated to the psycholinguistics approach of irony analysis. In psycholinguistics, researchers study the *processing time* of ironic utterances compared to the literal meaning (Gibbs Jr, 1993; Grice et al., 1975; Giora et al., 1998). Instead, here, we analyze the strategies adopted by the users to interpret the intended meaning of ironic messages. We discuss the psycholinguistics studies in detail in the related area sections of the following chapters.

I_{im}	H^1_{int}	H^2_{int}	H^3_{int}
It's encouraging how Portland Police Officers feel they're above the law	I'm discouraged that many Portland Police Officers think they're above the law	It's not encouraging how Portland Police Officers feel they're above the law	It's discouraging how Portland Police Officers feel they're above the law

Table 1.3: Example of speaker's ironic messages (I_{im}) and interpretations given by 3 Turkers (H^i_{int}).

1.1.3 Role of Irony to Detect Dis(agreement) Relations

Finally, in Chapter 5, we discuss the role of irony in argument mining research. Online discussion forums, blogs, webpage comments provide a wealth of naturally occurring arguments. Determining the nature of arguments and their relations is instrumental for argument mining and persuasion analysis research. A particular task has been to identify the agreement and disagreement in online discussion forums (for brevity, we adopted the naming convention of "dis(agreement)" from Rosenthal and McKeown (2015) to represent both relations together). We hypothesize that detection of irony can assist in identifying the dis(agreement) relations. We select a subcorpus from the Internet Argument Corpus (Walker et al., 2012b) where each response post is labeled with its argument relation (e.g., disagreement, agreement or neither relation to the quote it is responding to). Beside gold labels of argument relations, this corpus also contains information regarding whether a response is ironic to the previous post. Table 1.4 presents a pair of argument components and the argument relation (disagree or agree).

Argument Relation	Verbal Irony	Turn Pairs
Disagree	Irony	**userA:** I read it in the text. **userB:** Well , as we can see from other recent posts, your reading comprehension is below high school level. Why do folks think lying enhances their POV ?
Agree	Irony	**userA:** Today, no informed creationist would deny natural selection. **userB:** Seeing how this was proposed over a century and a half ago by Darwin, what took the creationists so long to catch up ?

Table 1.4: Ironic messages from userB and their respective prior turns from IAC.

Note, both response posts from UserB to UserA are ironic. We first exploit argument features to identify the argument relations from the pairs (results are comparable to the state-of-the-arts). Next, we demonstrate the impact of the irony features (features that are traditionally used to identify verbal irony and sarcasm) to detect the agree/disagree relation (we observe a 5% F1 improvement).

Before presenting the main contributions of the dissertation, we first discuss the definition and some characteristics of irony.

1.2 Irony and its Characteristics

The term *irony* takes its name from a character *Eirõn* featured in ancient Greek comedy (Kreuz and Roberts, 1993). *Eirõn* articulates beliefs and opinions that do not hold in order to conceal their actual feeling. This term also means the act of concealing the truth. According to the Merriam Webster (MW) dictionary, irony is defined as "the use of words to express something other than and especially the opposite of the literal meaning." In another definition in the same dictionary, it is noted that irony is often used in order to be funny. The meaning of irony has expanded to include at least four types of distinct concepts: *Socratic irony*, *dramatic irony*, *irony of fate*, and *verbal irony*. Kreuz and Roberts (1993) noted that although other types of irony are also possible (i.e., tragic irony), these four types are the basic descriptor of irony: all four share a common feature that there is a *discrepancy* between mental representation and the state of affairs. Below is a short description of the four types of irony.

- Socratic irony: This refers to the rhetorical technique of pretending ignorance to reveal a flaw in others' thinking.[4] An imperative characteristic of socratic irony is the act of *pretense*. The speaker knows the answer of the question but they are playing the role of the fool by acting as if they are not aware of the answer.
- Dramatic irony: This type of irony refers to a type of dramatic act: the audience possesses some information whereas the characters in the drama do not.
- Irony of fate: This is used by the speakers/authors to call attention to a particular incongruous relationship between two events. Kreuz and Roberts (1993) observed that there has been a tendency to overuse irony of fate to refer any odd or unexpected events ("a conference on electrical machines is having electricity problem!"). Irony of fate is also regarded as *situational irony*.
- Verbal irony: Here, the speakers/authors intentionally make statements that are opposite to the literal meaning. The utterances "Oh man, do I love doing sample returns" and "yeah, I really wanna be on public transportation ALL DAY. sounds GREAT!", are two examples of verbal irony, using positive words and phrases ("love," "great") but expressing negative sentiment towards the events of returning samples and spending all day on public transportation.

This conceptual variation between the different types of irony is one of the reasons why irony is approached differently from areas within philosophy, rhetoric, linguistics, and psychology (Regel, 2009). In this dissertation, I focus specifically on *verbal irony*.

Verbal irony is also defined similarly as irony, where the speakers/authors intentionally make statements that are opposite to the literal meaning. Note, although *sarcasm* is frequently treated as verbal irony, MW dictionary mentions that sarcasm is a subtype of verbal irony, which is used to insult or mock someone and to show irritation. The majority of the current supervised research in irony and sarcasm detection are trained on social media data (i.e., Twitter data) collected based on the hashtags (e.g., #irony, #sarcastic, #sarcasm) labels given by the authors. Naturally, it is unclear whether the

[4] The source of this type of irony refers to Socrates, the famous Greek philosopher who was known for his probing questions.

authors are clear about the subtle differences between these figurative use of language. For example, we observe that majority of the instances on Twitter that are labeled with the hashtag #sarcasm are instances of verbal irony and not sarcastic since the % of insults to others is low on Twitter. Instead, on Twitter, sarcastic tweets often intended to be funny and complain about the author's status (e.g., going to the emergency room during the weekend) and state of mind (e.g., being alone on Friday night; only one follower on Twitter). In contrary, in discussion forums, such as the Internet Argument Corpus and Reddit, we observe a decent number of sarcastic utterances that show irritation or mocking of others. These posts are mostly collected from controversial discussion forums on politics, religion, gun-control debate, etc. Since users often exchange heated arguments in such forums, it is not surprising that often these posts are bitter and sarcastic. Although there have been some empirical research efforts, notably (Attardo et al., 2003; Joshi et al., 2016) that propose differences in verbal irony and sarcasm, a clear distinction between these two phenomenons has not been established so far. In this thesis, we consider verbal irony/sarcasm as two interchangeable phenomena and use the working definition of verbal irony (i.e., use of words which actually means the opposite of what it seems to say [. . .]). Also, sometimes for brevity, we call verbal irony as simply *irony*.

The contributions, as well as the structure of the thesis is described next.

1.3 Contributions

Theoretically grounded computational methods: In this dissertation, we develop computational models to detect verbal irony that are grounded in theoretical frameworks from Linguistics, Philosophy, and Communication Science. We propose both machine learning methods based on discrete linguistically-motivated features and deep learning models that do not require feature engineering. We investigate *irony markers* and *irony factors*, two fundamental characteristics of irony (Burgers et al., 2012; Attardo, 2000b; Camp, 2012) (Chapter 3). We propose a reframing of verbal irony detection as a word-sense disambiguating problem: given an utterance, and a target word, identify whether the sense of the target word is literal or ironic. For this we propose a new Support Vector Machine kernel that uses word-embeddings, which is able to achieve the best performance even for target words where we do not have a lot of training examples. One of the irony factors that we model in this thesis is the incongruence between the literal meaning of irony and context (e.g., the incongruence between the expressed positive sentiment and the negative situation). This incongruence can happen inside an utterance, or between an utterance and the prior conversation context. We propose dual deep learning architectures that are able to model directly this incongruence.

Apart from irony identification, Chapter 4 is based on developing a new typology of linguistic strategies to categorize users' interpretation of ironic messages. Our research on irony interpretation complements behavioral research that looks at processing mechanism (e.g., the time taken to process ironic vs. literal utterances) of irony (Regel, 2009). Finally, Chapter 5 offers a new insight of the use of figurative language such as verbal irony in argument mining. Verbal irony can be used to express implicitly agreement or disagreement in online conversations. We investigate whether linguistic features used to detect ironic messages can help in detecting agreement and disagreement relations in online discussions.

User studies for explaining the prediction of models and for irony interpretation: One question that this dissertation investigates is "what triggers" an ironic reply in a conversation. We conduct user studies to address this question. We compare computational models with human annotations to investigate their agreement in identifying what part of the context can trigger an ironic reply in a conversation. This study is related to the question of explainability in computational (i.e., machine learning) models that are confirmed by user study. We looked at the important features (i.e., part of context) predicted by the models and investigated whether humans also recognize the same context that may trigger the ironic reply. We also examine if an ironic post contains multiple sentences, whether computational models and human annotations identify the same ironic sentences from the post.

In the second user study, we analyze how people interpret irony. We asked the annotators to rephrase ironic utterances to represent the intended meaning. We developed a new typology of linguistic strategies and investigate annotators' behaviors (i.e., for which ironic messages all the annotators use identical strategy to rephrase, etc.).

Research on diverse dataset: An essential aspect of supervised research is the availability of labels for the classification task. We look at two types of labels in this dissertation. One, self-annotated, where the users self-labeled their posts. For example, twitter dataset is *self-annotated* since the hashtags (e.g., "#sarcasm", "#irony") given by the authors are used as gold labels. The second type is *crowdsourced* (e.g., discussion forum data from the Internet Argument Corpus), where crowdsourcing is employed to annotate whether an online post is ironic or not. In the following chapters, we describe our research in detail.

We also released several datasets for the research community. Below is a short description of the datasets.

- Twitter data with target words: As part of our sense-disambiguation research, we built a corpus of close to 700K tweets collected based on sentiment words (e.g., "love", "great", "mature"; details in Section 3.5.2.1) balanced between irony/sarcasm and non-irony data. This dataset is available here.[5][6]

- Twitter data with time-stamp: For our research on evaluating the generalization quality of tweets we also built a corpus of approximately 660K tweets that come with time-stamp information. This data spans over tweets from 2013-2014 and 2014-2015.

- Twitter data with conversation context: We utilize a corpus of tweets in conversation thread for our research on our research on the role of conversation context in irony detection. This corpus contains around 26K tweets with full conversation (i.e., most of the conversation thread includes three-five tweets as context).

- Twitter data with rephrasing: This is the dataset used in irony interpretation research described in Chapter 4. This data contains 1,000 English, and 800 Spanish

[5] https://github.com/debanjanghosh/sarcasm_wsd

[6] According to the data sharing policy of Twitter, we are only allowed to share the serial numbers of tweets and not the actual text. This is inconvenient, because users often remove their tweets or delete their accounts.

tweets with five rephrase (i.e., rephrase present the intended meaning of the utterances) created by annotators.

1.4 Structure of the Chapters

- Data and Methods: Chapter 2 describes the data utilized in this dissertation. We introduce all the different corpora from different platforms, such as Twitter, Reddit, and Internet Argument Corpus. Our research is based on two types of social media content: self-annotated (i.e., speakers labeled their own utterances as ironic) and externally-annotated (i.e., external annotators annotated the utterances).

- Verbal Irony Identification: Chapter 3 presents various empirical models to identify verbal irony/sarcasm from social media. This chapter contains bulk of the research conducted in the dissertation. We develop *theoretically grounded computational methods* to study different characteristics of irony. We first introduce our study on the irony markers, the meta-communicative markers that alert readers about irony in utterances. We analyze the markers in microblogging platform as well as in the discussion forums (Section 3.5.1). Next, we discuss our research on irony factors. We first analyze irony detection as a sense-disambiguation problem (i.e., also referred as the "reverse of valence"; for details see section 3.5.2.1). We also investigate the semantic incongruence aspect of irony (see section 3.5.2.2). We also examine the role of conversation context in irony detection (section 3.6). We finish the chapter with an extensive qualitative study that addresses two questions. (1) can humans and computational models identify what part of a conversation context trigger an ironic reply, and (2) given an ironic utterance that contains multiple sentences, can humans and computational models identify the specific sentence that is ironic?

- Verbal Irony Interpretation: In chapter 4, we propose user studies to investigate how *readers interpret* verbal irony. We first conduct an annotation study where

Turkers on the Amazon Mechanical Turk (MTurk) platform rephrase ironic utterances to present the intended meanings. Next, we propose a new typology of linguistic strategies that Turkers have utilized to interpret irony. We empirically validate the linguistic strategies over two datasets to automatically identify the strategies. We conduct thorough analysis of the behaviors of the Turkers. We particularly analyze the annotations of three Turkers who finished the most number of tasks (i.e., around five-hundred rephrasing) to detect where they use similar and dissimilar strategies for interpretation. Further, we provide an analysis of particular ironic utterances, i.e., utterances for which all Turkers use similar strategy for interpretation, and where they all took different strategies.

- Role of Verbal Irony for Dis(agreement) Detection: Chapter 5 demonstrates the use of irony recognition for down-stream applications. For instance, we investigate the role of verbal irony, particularly, in detecting dis(agreement) relations between online posts in discussion forum. We use the IAC corpus where posts are labeled with dis(agreement) relations as well as with verbal irony labels. We utilize state-of-the-arts features for both tasks, dis(agreement) detection as well as irony detection. We show how irony features assist in identifying dis(agreement) relations.

- Conclusions and Future Work: finally, in Chapter 6 we summarize our contributions of this dissertation. We also discuss the limitations and future work.

Chapter 2
Data

2.1 Overview

In this chapter, we introduce all the datasets used in this dissertation. Two platforms of social media data are used in our research: micro-blogging platform such as Twitter and online discussion forums such as $Reddit$ and Internet Argument Corpus (for brevity, henceforth IAC). We briefly introduce the two platforms here.

With the rapid development of social media, spontaneously user-generated content is extremely important to analyze users' opinions and sentiments online. Out of all the social media platforms, Twitter has become a major resource for research in natural language processing, from sentiment analysis (Agarwal et al., 2011; Bollen et al., 2011; Rosenthal et al., 2017) to event and relation detection (Becker et al., 2011; Weng and Lee, 2011; Ritter et al., 2012), topic modeling (Ramage et al., 2010; Hong and Davison, 2010), and dialog acts identification (Ritter et al., 2010). Twitter has also been a major source of information for different applications such as earthquake detection (Sakaki et al., 2010) and disease surveillance (Lamb et al., 2013). One of the major characteristics of Twitter is it allows its users to write messages up to only 140 characters.[1] Since the messages are short, users use various indicators, such as hashtags, URLs, and emoticons. Hashtags (#hashtag) are tags that are often assigned by the author to mark content/topic/category (e.g., #teaparty, #worldcup), sentiment (e.g., #angry, #sad, #happy, #sarcasm), and/or location (e.g., #Paris, #NYC), among other uses. These hasthags are used for indexing purposes; tweets that contain a candidate hashtag will be retrieved if someone searches in Twitter using the hashtag as a query. For instance the

[1] Recently Twitter has doubled the length of the messages to 280 characters. However, majority of our research is based on tweets that are collected between 2013-2016 so our training dataset is still based on the tweets that are up to 140 characters.

tweet, “each morning presents a new beginning #love #compassion URL”, contains two hashtags #love and #compassion. Next, if any of the two hashtags are queried this particular tweet can be retrieved from Twitter. This is one of the main motives of users to label tweets with sentiment hashtags. From the perspective of NLP research, the length restriction is both an advantage (lexical factors may be more prominent than syntactic factors) and a challenge (abbreviations and symbols with special interpretation in Twitter may decrease the effectiveness of natural language processing tools optimized for more standard language use). Data collection from Twitter for different experiments are described in Section 2.2.1.

Apart from Twitter we also analyze social media platform such as online discussion forums. An increasing portion of information and opinion exchange occurs in online interactions such as discussion forums, blogs, and webpage comments. Online posts in discussion forums such as $Reddit$ and IAC contain many instances of verbal irony and sarcasm.

$Reddit$ is a social news aggregation, web content rating, and discussion website. Members of the website can posts messages, links, as well as images. Posts are organized by subject into user-created discussions called “subreddits”, which cover a variety of topics including news, science, movies, video games, music, books, fitness, food, and image-sharing. Specific subreddits of $Reddit$ such as the “change-my-view” subreddit has already became popular for research on automatic detection of persuasion, influence, and concessions (Tan et al., 2016a; Hidey and McKeown, 2018; Musi, 2017). IAC is also based on discussions extracted from the online debate site 4forums.com, ConvinceMe, and CreateDebate (Abbott et al., 2016). This corpus is used in multiple research related to argument mining such as dis(agreement) detection (Misra and Walker, 2013; Abbott et al., 2011). Since many subreddits in $Reddit$ and discussion threads in IAC are based on contentious topics we observe users often reply with ironic and sarcastic posts in the discussion.

One major difference between Twitter and the discussion forums is naturally the length of the posts. As stated earlier, Twitter limits users to write up to 140 characters where as discussion forums do not have such limitation. We observe, often ironic posts in $Reddit$ and IAC contain multiple sentences. For instance, the majority of the posts in IAC are between three and ten sentences.

Another difference among the social media platforms stems from the nature of the labels. Supervised classification models use gold labels for training and we observe two types of labels are available here. (a) data that are self-annotated with sarcasm or verbal irony labels, and (b) data that are annotated by external annotators such as crowdsourcing. For instance, Twitter and $Reddit$ dataset can be denoted as *self-annotated* corpus since we have used the hashtags (e.g., "#sarcasm", "#irony") as labels whereas discussion forum data from the Internet Argument Corpus (IAC) is *external-annotated* since crowdsourcing is used to annotate the posts. In other words, for Twitter and $Reddit$ corpus our models are based on how users/author perceive the concept of verbal irony since we are directly using their labels. In contrary, for IAC, annotators were provided with the definition and examples of sarcasm while annotating.

All three corpora are described in the following sections.

2.2 Datasets

2.2.1 Twitter Dataset

As stated earlier, we have relied upon the annotations that users assign to their tweets using hashtags. For instance, users label sarcastic and ironic tweets with hashtags such as #sarcasm, #sarcastic, or #ironic. We used Twitter developer APIs to collect tweets for our research.[2]

To identify ironic tweets we focused on classification models (i.e., we train verbal ironic utterances vs. non-ironic utterances). As non-ironic utterances, we consider random objective utterances (i.e., tweets that do not have sentiment, irony, or sarcasm related hashtags) as well as sentiment tweets. These sentiment tweets are composed of positive and negative sentiment. The positive tweets express direct positive sentiment and they are collected based on tweets with positive hashtags; #happy, #love,

[2] Particularly, we use two libraries, the "twitter4j" (in Java) and the "twarc" (in Python) to accumulate the tweets.[3]

Category	Utterance
I	. . . starting off the new year great !!!!! sick in bed . . .
I	yay something to be proud of 3rd poorest in the NATION . . .
NI_{rand}	. . . you don't need a record label to have great music . . .
NI_{rand}	im filipino with dark brown eye and forever true and proud . . .
NI_{sent}	. . . i'm in love with this song great job justin . . .
NI_{sent}	but i'm proud of all the beliebers AROUND THE WORLD . . .

Table 2.1: Examples of ironic and non ironic utterances

#lucky, etc. Similarly, the negative tweets express direct negative sentiment and are collected based on tweets with negative hashtags; #sadness, #angry, #frustrated etc. Although most of the related research on verbal irony detection identifies ironic utterances against "any" non-ironic utterances (i.e., objective utterances) we use both random objective and sentiment utterances for classification. We reckon classifying ironic against sentiment utterances is a difficult task since many ironic utterances also contain sentiment terms. See Table 2.2 for a full list of hashtags used in our research. For brevity, we denote ironic utterances as I and non-ironic utterances as NI. The non-ironic objective utterances are denoted as NI_{rand} and the sentiment (non-ironic) utterances are NI_{sent}. Table 2.1 presents examples of ironic, non-ironic sentiment, as well as non-ironic random utterances. The first column in the table depicts the type of the tweet (whether it is verbal irony or not) and the second column presents the utterance. Table 2.3 represent the overall number of the tweets used in different experiments in this dissertation.

We utilize tweets that are individual and irrespective of any conversation as well as tweets that are part of a conversation. Conversation tweets are usually marked by "$@ < user >$", where the interpretation of a particular tweet is most likely dependent on the entire conversation. based on the occurrence of "$@ < user >$" symbol we collected the full dialogue (whenever available) and used in our research described in Chapter 3.

We utilize a couple of preprocessing techniques that are only applied to tweets. For instance, we lowercased the tweets, except the words where all the characters are uppercased (e.g., we did not lowercase "GREAT", "SO", and "WONDERFUL" in the

Category	Hashtags for CollectingTweets
Irony (I)	#irony, #sarcasm, #sarcastic
Positive (P)	#happy, #joy, #happiness, #love, #grateful, #optimistic, #loved, #excited, #positive, #wonderful, #positivity, #lucky
Negative (N)	#angry, #frustrated, #sad, #scared, #awful, #frustration, #disappointed, #fear, #sadness, #hate, #stressed

Table 2.2: Sentiment Hashtags for Collecting Tweets

Experiment	Data used	Comment
sarcasm/verbal irony markers	350K utterances	(Section 3.5.1)
generalization of markers	660K utterances	timestamp of the utterances are used to split utterances based on weeks/months for experiments (Section 3.5.1.2)
verbal irony sense-disambiguation	total 712K utterances	separate experiment on target words. Size of training data varies from 56K (target word "love") to 1K (target work "mature") (Section 3.5.2.1)
co-text incongruence identification	- same -	- same - (Section 3.5.2.2)
verbal irony identification using conversation context	26K utterances	data is collected based on whether tweets are replies to other tweets; we collect the full thread of dialogue (Section 3.6)
irony interpretation research	1K utterances and 5K rephrases from annotators	Rephrases represent the intended meaning of the users (Chapter 4)

Table 2.3: Tweet Datasets for different irony identification and interpretation research

utterance, “GREAT i’m SO happy shattered phone on this WONDERFUL day!!!”. We deleted all tweets where the hashtags of interest were not located at the very end of the message. As a result, we eliminated utterance such as “#sarcasm is something that I love”. We removed the retweets (e.g., tweets that start with “RT”), duplicates, and quotes. Finally, we also eliminated tweets written in languages other than English using the library Textblob.[4]

2.2.2 Reddit Corpus

Khodak et al. (2017) introduce the self-annotated Reddit Corpus, which is a very large collection of sarcastic and non sarcastic posts (over one million) from different subreddits. This corpus contains the prior turn that is either the original post or a prior turn in the discussion thread that the current turn is a reply to. This corpus contain self-annotated data, that is, the speakers labeled their own posts/comments as sarcastic using the marker “/s” to the end of sarcastic posts. For obvious reasons, the data is noisy since many users do not make use of the marker “/s”, do not know about it, or only use it where sarcastic intent is not otherwise obvious. Khodak et al. (2017) have conducted a evaluation of the data having three human evaluators manually check a random subset of 500 comments from the corpus tagged as sarcastic and 500 tagged as non-sarcastic, with full access to the post’s context. They found around 3% of the non-sarcastic data is false negative. We collected 50K posts (balanced between verbal irony and non-verbal irony) and denote this corpus in subsequent section as $Reddit$. Table 2.4 shows an example of sarcastic current turn (userF’s post) and its prior turn (userE’s post) from the $Reddit$ dataset.

We use standard preprocessing, such as sentence boundary detection and word tokenization when necessary. Also, we selected posts as well as their contexts only when the reply posts are at least two sentences and their context are at least three (maximum of seven) sentences. These thresholds are used to determine whether our algorithms can detect (a) sentences from the context that trigger verbal irony in replies and (b) handle posts that are longer (i.e., more than one sentence). This corpus is used in two

[4] Textblob:http://textblob.readthedocs.io/en/dev/

Turn pairs
userA: nothing will happen, this is going to die a quiet death like 99.99 % of other private member motions. this whole thing is being made into a big ordeal by those that either don't know how our parliament works, or are trying to push an agenda. feel free to let your mp know how you feel though but i doubt this motion gets more than a few minutes of discussion before it is send to the trashcan. **userB:** the usual "nothing to see here" response. whew! we can sleep at night and ignore this.
userA: They're (media) lying. The Uk, covers up islamic violence. Just pretend it's not happening is the rule. Any who points out the truth will be arrested. **userB:** But did they mention the gender of the attacker? No, because the media is covering the fact that it is 99of the times men that commit these crimes. #deportallmen.

Table 2.4: Examples from *Reddit*. We show sarcastic posts (posts from userB) as well as their conversation context (posts from userA)

studies. (1) examining the use of irony markers in *Reddit* threads and (2) the role of conversation context in irony detection.

Note, Wallace et al. (2014) first suggested using the Reddit discussion forums for irony detection research. They released a small corpus of Reddit posts where each sentence in every post had been labeled by three independent annotators. However, the corpus used in their research was small and heavily skewed - only 292 posts contain an instance of verbal irony where the rest of 1,153 posts do not. Thus this corpus is not suited for empirical research that we have conducted in the following chapters.

2.2.3 Internet Argument Corpus

Apart from self-labeled utterances we also use verbal ironic utterances that are labeled by external annotators. For instance, we use discussion forum data from the Internet Argument Corpus (*IAC*) (Walker et al., 2012b). Internet Argument Corpus (*IAC*) is a publicly available corpus of online forum conversations on a range of social and political topics from gun control debates, marijuana legalization, climate change, evolution, and so on. *IAC* comes with annotations of different types of social language categories such as agreement/disagreement (between a pair of posts), nastiness, and sarcasm. There are different versions of *IAC* and we use two such subsets of *IAC* in our research.

First, we utilize, the “Sarcasm Corpus 2”, a corpus that is annotated by a large number of annotators (i.e., 5 to 7 Turkers) to identify sarcasm (Justo et al., 2014; Oraby et al., 2016a). Oraby et al. (2016a) have introduced this corpus, which is a subset of the Internet Argument Corpus V2. This corpus (denoted as IAC_{v2} in our research) contain 9,400 posts labeled as sarcastic or non-sarcastic (balanced dataset). IAC_{v2} not only contains sarcastic posts but all the contexts (i.e., prior posts or prior turns) to which the sarcastic posts are replies to. To obtain the gold labels, Oraby et al. (2016a) first employed a weakly supervised pattern learner to learn sarcastic and non-sarcastic patterns from the IAC posts and later employ a multiple stage crowdsourcing process via Amazon Mechanical Trunk. Although the dataset described by Oraby et al. (2016a) consists of 9,400 post, only 50% of that corpus is currently available for research (4,692 altogether; balanced between sarcastic and non-sarcastic categories). This is the dataset we used in our study, particularly, for the research on examining the role of conversation context (Section 3.6).[5] Table 2.5 show examples of ironic current turn (userB’s post) and its prior turn (userA’s post) from the IAC_{v2} dataset.

Beside the IAC_{v2} we also extract a subset of posts from IAC that we utilize in our research on investigating the role of verbal irony to determine the argumentative relations. We collected around 10K posts as well as their context that are marked with argumentative relations (i.e., agree/disagree/no-relation) and sarcasm (i.e., sarcastic or non-sarcastic posts).

Apart from the nature of annotators, there is another difference between the two types of the corpus. On one hand, for the *other-annotated* corpus, the annotators are usually provided with proper guidelines to identify verbal irony or sarcasm in discussion forums. For instance, (Oraby et al., 2016a) provided two definitions and a couple of examples of verbal irony. The first definition states that sarcasm is a sharp and often satirical phenomena designed to be snarky, mocking, or humorous and the second definition states that sarcasm is a mode of satirical wit often directed towards an individual

[5] Oraby et al. (2016a) reported best F1 scores between 65% to 74% for sarcasm detection. However, we foresee the reduction in the training size of the available corpus will have obvious effects in the classification performance.

Turn pairs
userA: How do we rationally explain these creatures existence so recently in our human history if they were extinct for millions of years? and if they were the imaginings of bronze age sheep herders as your atheists/evolutionists would have you believe, then how did these ignorant people describe creatures we can now recognize from fossil evidence? and while your at it, ask yourself if it's reasonable that the bones of dead creatures have survived from 60 million years to some estimated to be more than 200 million years without becoming dust? **userB:** How about this explanation - you're reading WAAAAAY too much into your precious Bible.
userA: If I may chime in, George W. Bush was not responsible for 9/11. As I have stated on one of the other threads, Bill Clinton,et. al. did not do a rat's (explicitive deleted) to improve intelligence sharing between Law enforcement and intelligence. Does Able Danger ring a bell? :-S :-S **userB:** Stick with the bass guitar.@@ And if you break a string....TRY not to blame it on Clinton. :P

Table 2.5: Examples from IAC corpus. We show ironic posts as well as their conversation context here.

or situation. On the other hand, it is unclear whether the authors of the self-labeled sarcastic utterances (i.e., on platforms such as Twitter or Reddit) follow any specific guideline or definition of verbal irony. It is highly plausible that the authors use their own interpretation while labeling verbal irony or sarcasm.

Table 2.6 presents the gist of all the experiments and data used. The first column shows the main three aspects (i.e., verbal irony identification, interpretation, and application) of the dissertation. The second column presents the source of the data. Finally, the third and the fourth column present the gist of the experiments. The subscript u denotes the experiments where "utterance" are used independently of the context c. $u + c$ subscripts depicts the experiments where the utterance u and the context c both are used.

Type of Experiments	Data source	Type of Data	Experiments
(1) irony identification	$Twitter$	$Twitter_u$	verbal irony markers
	$Twitter$	$Twitter_u$	verbal irony factors (reversal of valence, context incongruity)
	$Twitter$	$Twitter_{u+c}$	role of conversation context
	$Twitter$	$Twitter_{u+c}$	role of conversation context
	IAC_{v2}	$IAC_{v2_{u+c}}$	role of conversation context
	$Reddit$	$Reddit_u$	verbal irony markers
	$Reddit$	$Reddit_{u+c}$	role of conversation context
(2) irony interpretation	$Twitter$	$Twitter_u$ -	
(3) role of irony	IAC_{role}	$IAC_{role_{u+c}}$	verbal irony in argumentation research

Table 2.6: Different corpus and associated experiments (u denotes utterances and c denotes context

Chapter 3
Verbal Irony Identification

3.1 Overview

Human communication often employs the use of verbal irony and sarcasm. [1] Failing to identify verbal irony will lead to errors in detecting people's attitude such as sentiment and beliefs. In this chapter, we discuss our research on building computational models for detecting verbal irony. There are three major contributions. First, we model various characteristics of verbal irony, such as irony factors and irony markers that are based on theoretical frameworks of irony. Second, we model utterances in isolation as well as part of a local conversation context. Third, we utilize a dual neural network architecture to model the contextual knowledge. For instance, in Section 3.5.2.2, we build a dual Convolutional Neural Network (CNN) where one CNN models the co-text (i.e., context that is present in the utterance) and the other CNN models the sentiment of the utterance. Likewise, in Section 3.6 our analysis is based on Long Short-Term Memory (LSTM) networks that models each message/turn in a social media conversation separately.

This chapter is structured as follows. First, Section 3.2 introduces the theoretical underpinnings of verbal irony and Section 3.2.1 discusses related studies on irony recognition from linguistics and psycholinguistics studies. In Section 3.2.2 we discuss related research in Natural Language Processing on automatic detection of verbal irony. Next, in Section 3.4, we briefly introduce the datasets and data representation used in this chapter.[2] Following that, we introduce our empirical investigations on verbal irony identification. We first consider the *utterances* in isolation (i.e., without any contextual

[1] Parts of the chapter already published in (Muresan et al., 2016; Ghosh et al., 2017, 2015; Ghosh and Muresan, 2018). Also, the section on *conversation context* is accepted (with minor revisions) at Computational Linguistics Journal.

[2] Note, Chapter 2 discusses all the datasets used in this dissertation in details.

knowledge) (Section 3.5). Second, we use the *conversation context*, a type of contextual information to discover whether the use of context assists in verbal irony identification (Section 3.6). As stated before, we use the terms — verbal irony and sarcasm – interchangeably.

3.2 Background and Related Work

Irony and sarcasm are well-studied phenomena in linguistics, philosophy, psychology, and communication, where several theoretical frameworks have been put forward, sometimes at odds with each other (Grice et al., 1975; Gibbs, 1986; Gibbs and Colston, 2007; Giora, 1995; Kreuz and Glucksberg, 1989; Utsumi, 2000; Camp, 2012). In the Natural Language Processing community, we observe a recent surge in research on automatic methods for sarcasm detection, mainly treating the problem as a binary classification task (González-Ibáñez et al., 2011; Riloff et al., 2013; Davidov et al., 2010; Liebrecht et al., 2013; Joshi et al., 2015; Reyes and Rosso, 2011). We first discuss some of the competing theories that attempt to explain irony and sarcasm and then consider the recent empirical studies.

3.2.1 Theoretical Approaches of Irony Analysis

In the introduction chapter we present different definitions and types of irony. We also discuss the scope of sarcasm and verbal irony and their similarities. Broadly speaking, we are following the typical definition of verbal irony that is often defined as a figurative language in which the speakers say the opposite of what they mean. We observe that this definition captures the most visible and prevalent cases of verbal irony and sarcasm. For instance, if John utters "I love going to the emergency room during the weekend", it is likely that John is being ironic and John expresses the exact opposite of the literal meaning (i.e., John hates to go to the emergency room during the weekend).

Grice et al. (1975) first put forward a discussion of figurative language and proposed four rules of conversation obeyed by the interlocutors (Grice et al., 1975; Grice, 1978). These rules are referred as "conversational maxims". They are:

- *The maxim of quantity*: where one tries to be as informative as one possibly can, and gives as much information as is needed, and no more.
- *The maxim of quality:* where one tries to be truthful, and does not give information that is false or that is not supported by evidence.
- *The maxim of relation:* where one tries to be relevant, and says things that are pertinent to the discussion.
- *The maxim of manner:* when one tries to be as clear, as brief, and as orderly as one can in what one says, and where one avoids obscurity and ambiguity.

Grice proposed the Standard Pragmatic Model (for brevity, SPM) that presumes interlocutors contribute to an efficient and successful conversational exchange by conveying truthful, relevant and clear information. However, figurative utterances such as irony are assumed to violate the principle of maxim, thus allowing for the possibility of *implicatures*. According to Grice, in the above example, John is intentionally violating the *maxim of quality* (Grice et al., 1975; Attardo, 2000b; Palinkas, 2013). In Grice's explanation, ironic expressions are untrue in their literal meaning and understanding of irony is a two-stage process. In the first stage, the literal meaning (i.e., John loves going to the emergency room) is evaluated, and when the reader finds that the literal meaning is incongruent with the context, it is rejected. Next, the intended meaning (i.e., John hates going to the emergency room during the weekend), which is the opposite of the literal meaning is initiated.

In contrary to SPM, Gibbs (1986) proposes the "direct access view" model that suggests instead of the two-stage processing for ironic messages, humans use identical mechanisms to process figurative and literal language. According to this view, the comprehension of figurative language does not involve any additional cognitive processes, but rather processing is largely based on the pragmatic knowledge used by the human to understand ironic messages.

Giora et al. (1998) came up with the "Graded Salience Hypothesis" that sits in between the "SPM" and "direct access view" model. It states that processing of lexical meaning (i.e., figurative or literal) is a graded process (i.e., not a binary process) that is heavily based on retrieving the *salient meaning* of words from the mental lexicon (Giora et al., 1998). When expressions have multiple meanings, the most salient meaning is first retrieved and then the remaining ones are graded. In contrast to the direct access view, contextual information does not play a very important role in accessing the salient meanings. Giora also proposed another model of irony interpretation that is based on the indirect negation view of irony. According to this model, the hearer retains the literal sense of the utterance to compute the difference between the literal meaning and the less desirable, ironic meaning. In this respect, the indirect negation view of irony differs from Grice's model (Grice et al., 1975), which assumes that the literal meaning should not be retained, but rejected and suppressed. This model also states that when one expresses irony, it is equivalent to adding a negation term like "not".

Wilson and Sperber (1992) critiques the SPM model and argues that the very definition of verbal irony (i.e., intended meaning is the opposite of the literal meaning) is not comprehensive. First, this definition does not capture the cases of *ironic understatements*. For instance, Sara's friend returns her suit with a large wine stain. In response, she made an understatement, "it doesnt look too bad". Here, the literal opposite will be that the suit looks great, however, this was *not* what the speaker intended. The utterance "it doesnt look too bad" mocks the proposition, i.e., given the circumstances of wine stains, it is absurd to make such a statement and it is exactly what the speaker is ironic about (Wallace, 2015). Wilson and Sperber (1992) provide additional examples of verbal irony that are not captured by the SPM and they discuss the model of *echoic mention*. In echoic mention, verbal irony implicitly suggest to some real or hypothetical proposition in the form of quoting opinions of other persons. Here, speakers merely repeating utterances made by others in order to achieve irony. Burgers (2010) discussed a particular example of *echoic mention*. Suppose, someone is visiting Tuscany in the summer to enjoy the great weather. However, they face a terrible storm and exclaim - "Ah, Tuscany in May" (Wilson and Sperber, 1992). Here, the speaker suggesting to some hypothetical proposition (i.e., enjoying weather during the storm, probably uttered by his local friends from Tuscany) to show the absurdity of the situation by mocking using irony. According to Wilson and Sperber (1992) it is hard, if not impossible to explain this type of irony through the SPM, because, it is unclear whether there is any particular lexical item in this type of utterance that can be replaced by its semantic opposite to identify the intended meaning.

Clark and Gerrig (1984) proposed the "pretense theory" where the authors categorize the irony audience in two categories. This theory suggests two type of audiences: those who can detect the ironic instances and understand the intended meaning and those who will accept the ironic utterances at its surface literal meaning.

Attardo (2000b) regards irony as a type of "relevant inappropriateness" that is equivalent to Grice's SPM. According to the "relevant inappropriateness" model, ironic utterances are contextually inappropriate and the readers reject the literal meaning, since they understand that the utterance is inappropriate, and accept the intended meaning. Attardo (2000b) also introduced two major characteristics of irony - *irony markers* and *irony factors*. In Section 3.3 we continue the discussion on irony factors and irony markers since we analyze these aspects in detail in this discussion.

3.2.2 Computational Approaches for Irony Identification

Supervised computational models for detecting irony and sarcasm depend on having access to annotated data. Since annotating and building a corpus for irony detection is expensive, most research relies on social media data, such as tweets because users label the tweets using hashtags. Hashtags such as #sarcasm and #sarcastic, in turn, are used as gold labels for supervised sarcasm detection. As we will see subsequently in this chapter and in related research, Twitter is the de-facto platform for collecting training data in verbal irony based research (Davidov et al., 2010; Reyes and Rosso, 2011; González-Ibáñez et al., 2011; Riloff et al., 2013; Joshi et al., 2015; Muresan et al., 2016; Wang et al., 2015; Bamman and Smith, 2015; Rajadesingan et al., 2015; Reyes et al., 2013). Apart from tweets, researchers have also used discussion forums such as Reddit discussion threads (Khodak et al., 2017; Wallace et al., 2014) and Internet Argumentative Corpus (for brevity henceforth, IAC), which consist of debate threads ranging from political and social topics topic such as "gun control" to "death penalty", etc. (Justo et al., 2014; Oraby et al., 2016a)

Method-wise we observe that researchers have utilized rules, discrete features (i.e., n-grams and lexicons) as well as word embedding and neural networks based techniques to identify verbal irony. We briefly discuss the approaches here. Regarding use of the hand-crafted rules, Veale and Hao (2010) came up with a pattern "* as a" and used Google search to determine whether such patterns are sarcastic. They categorize patterns such as "as **private** as a **park-bench**" or "**crazy** as a **fox**" as sarcastic. They employed different types of post-processing based on the web frequency of the pattern and lexical/morphological similarities between the **bold** words. For instance, "as **cool** as a **cucumber**" or "**manly** as a **man**" are not considered sarcastic due to the high lexical similarities between **cool** and **cucumber** and **manly** and **man**. However, the scope of finding verbal irony with only such a single pattern is highly limited. Instead of hand-crafted text patterns, Maynard and Greenwood (2014) studied the use of various hashtags to identify sarcasm in Twitter. They have developed a set of rules based on the "polarity" of the hashtags to identify sarcasm. For example, in the utterance "heading to the dentist. #great #notreally", they first identify that "#notreally" is a irony indicator (Burgers et al., 2012) and it is following a positive hashtag "#great". In cases where an irony indicator follows another hashtag (that is presenting positive or neutral sentiment) Maynard and Greenwood (2014) alter the sentiment of the tweet and identify sarcasm. Similar to Maynard and Greenwood (2014), Liebrecht et al. (2013) looked into the use of the particular hashtag "#not" as label for verbal irony detection. In a similar vein, (Davidov et al., 2010; Tsur et al., 2010) aim to identify sarcastic utterances from Twitter and Amazon product reviews using semi-supervised approach (i.e., textual patterns) as well as syntactic features. The semi-supervised strategy exploits labeled as well as unlabeled instances. Their patterns are based on word frequencies (i.e., use of frequent words and content words) whereas the syntactic features capture the number of quotations, exclamation marks, uppercase words etc. Such features are common in verbal irony detection and we also have used such features in our research. Davidov et al. (2010) and Reyes and Rosso (2011) have tackled the problem of sarcasm detection in customer reviews on Amazon, comparing the sarcastic reviews to plain negative reviews from Amazon and Slashdot.

Beside the pattern based semi-supervised approaches the majority of the verbal irony detection research is fully supervised. Although most of the classifiers classify ironic against "any" non-ironic utterances (i.e., they could be objective without any sentiment), González-Ibáñez et al. (2011) considered the somewhat harder problem of distinguishing ironic tweets from non-ironic tweets that directly convey positive and negative sentiment. They used lexical features (i.e., n-grams) and lexicons such as Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2001a) and WordNet Affect (Strapparava et al., 2004). In addition, their work explored the use of indicators and pragmatic features such as emoticons, interjection, and the use of "@user" feature (also examined by Carvalho et al. (2009)).

We also observe a few recent research attempts to detect verbal irony in languages other than English. For example, Ptáček et al. (2014) use various n-grams, including unigrams, bigrams, trigrams and a set of language-independent features, including punctuation marks, emoticons, quotes, capitalized words, character n-grams features to identify sarcasm in Czech tweets. Similarly, (Liu et al., 2014) introduce POS sequences, homophony features to detect sarcasm from Chinese utterances. Bharti et al. (2017) compared tweets written in Hindi to news context for irony identification.

Similar to other NLP problems such as machine translation, parsing, relation extraction, textual entailment (Bahdanau et al., 2014; Chen and Manning, 2014; Zeng et al., 2014; Rocktäschel et al., 2015) to name a few, recently researchers are also involved in using word embeddings and deep learning based techniques to detect verbal irony. Zhang et al. (2016) employed a bi-directional gated recurrent neural network to categorize sarcastic tweets. Their results show that neural network based models give improved accuracies for verbal irony detection compared to baselines (i.e., based on content word and historical tweets of the authors). Ghosh and Veale (2016) used a combination of the convolutional neural network (CNN) and the Long Short-Term Memory networks (LSTM) and achieved over 90% accuracy when they trained their model on tweets.

Apart from looking only at lexical/pragmatic features for the binary classification task, researchers have also investigated different properties of verbal irony. For instance, Riloff et al. (2013) observed that a common form of verbal irony on Twitter consists of a positive sentiment contrasted with a negative situation. For example, many ironic tweets include a positive sentiment, such as "love" or "enjoy", followed by an expression that describes an undesirable activity or state (e.g., "taking exams" or "being ignored"). They presented a bootstrapping approach that identifies such positive sentiment words and and negative situations. Joshi et al. (2015) improved the recall by allowing negative seed words to this bootstrapping approach.

The effect of contextual information is also studied in verbal irony detection, in the form of modeling the conversation context (Wallace et al., 2014) or modeling the author (i.e., previous tweets or previous posts) (Bamman and Smith, 2015; Amir et al., 2016; Khattri et al., 2015). Khattri et al. (2015) studied the previous tweets of a user to identify whether the sentiment expressed towards an entity in the candidate tweet agrees with the sentiment expressed by the author towards that same entity in the past tweets. Checking the author's previous tweets can confirm (a) if an user is often ironic to a specific target, and (b) if the sentiment alters (i.e., negative to positive) towards the target, which may results in verbal irony. Amir et al. (2016) created user embeddings based on the tweets of users and combined that with regular utterance-based word embeddings to show improvement in irony detection. Instead of learning the history of the authors (Bamman and Smith, 2015) used only the previous tweet (if a sarcastic tweet is a part of a conversation) to build a local-context based model that showed a modest improvement in verbal irony detection accuracy.

Beside the linguistically motivated contextual knowledge, cognitive features, such as eye-tracking information is also used in irony detection (Mishra et al., 2016). Tepperman et al. (2006) used a particular audio feature where individuals uttered the phrase "yeah right" both ironically and literally. They encoded various spectral features such as pitch of voice, rising/falling frames etc. They achieved F1 of 70%. Similarly, Scharrer and Christmann (2011) studied how voice modulations accompany ironic utterances in speech. Schifanella et al. (2016) proposed a multi-modal approach, where textual and visual features are combined for irony detection.

Although all of these recent studies are pushing the boundary of verbal irony detection in social media and other genres, our focus is on *theoretically grounded computational models* of irony analysis in the text. We model two characteristics of verbal irony, irony markers and irony factors that shows analysis of such characteristics helps in irony detection. We propose linguistically motivated models as well as deep learning models. We model utterances both in isolation and when part of a conversation context.

3.3 Introduction to Irony Markers and Irony Factors

Attardo (2000b) proposed a theory of irony and discussed two particular characteristics of ironic utterances that we also observe applicable to written communication expressing verbal irony and sarcasm. These two characteristics are *irony markers* and *irony factors*, and we adopt them in our analysis. *Irony markers* are explicit indicators of irony. They are the "meta-communicative" clues that alert a reader about the presence of irony in an utterance (Attardo, 2000b; Burgers et al., 2012). In contrast, irony markers can be removed without destroying the irony. For example, the capitalization of "NEVER" in the ironic utterance "A shooting in Oakland? That NEVER happens", signals verbal irony, but removing the marker, i.e., the capitalization of the word "NEVER" to "never" will not affect the presence of irony. Attardo (2000b) also noted that these markers in written communication behave like indicators of verbal irony in face-to-face conversation (e.g., facial expression, posture, voice intonation). In contrary, an *irony factor* is an inherent characteristic of the ironic utterance. This means that an irony factor cannot be eliminated from a ironic utterance without destroying the irony. For instance, consider the utterance that we introduced in the previous section (e.g., "I love going to the emergency room during the weekend"). We can modify the utterance in two ways. First, we can alter the *situation* (e.g., "I love going to the soccer field during the weekend"). Second, we can modify the *sentiment* (e.g., "I hate going to the emergency room during the weekend"). Both changes remove irony from the original utterance.

3.3.1 Theoretical Overview of Irony Markers

In spoken communication, there are various markers used by the speakers to express irony and sarcasm, namely, the variation of the *intonation*, *nasalization*, and *excessive stress* (Attardo, 2000b). However, while writing, users cannot use these verbal markers and instead they use capitalization to express stress, and interjection, emoticons and emojis to communicate different types of intonations. An irony marker should be interpreted as a "clue" to help a reader to understand an utterance is ironic. In other words, for instance, if an ironic utterance contains hyperbolic words and multiple exclamation marks then it could be easier to recognize irony in it. However, this does not imply that use of a hyperbolic word always leads a reader to a ironic interpretation. Rather, this serves as a clue to indicate verbal irony (Burgers, 2010).

Hallmann et al. (2016) mentioned that historically, in literature and poems, irony markers were never preferred, particularly, due to the reason that markers being explicit "meta-communicator" would be a spoiler and ambiguity is precisely one of the goals of poets. However, in social media, it is common to use markers because the ironic intention of the author may go unnoticed. Moreover, tweets are short messages of only one-hundred and forty characters. Burgers (2010) categorize the markers in three major categories – tropes, morpho-syntactic, and typographic. We discuss them in the following section.

3.3.1.1 Tropes as Irony Markers:

Tropes are a type of figurative language. An ironic instance can be considered a trope in itself.

- Metaphors - Metaphors often facilitate ironic representation and are used as markers. Metaphors are defined as "implicit comparisons" between two entities or mentions where the entities are coming from two domains.

- Hyperbole - Hyperboles or intensifiers are commonly used in irony because speakers frequently overstate the magnitude of a situation or event. For instance, words that are denoted as “strong subjective” (positive/negative), such as “greatest”, “best”, “genius”, are frequently used in ironic utterances to stress the intensity.

- Rhetorical Questions - Rhetorical questions have the structure of a question but are not typical information seeking questions since the reader is not supposed to provide an answer to the question. Rhetorical questions are very common markers for verbal irony, especially in social media.

3.3.1.2 Morpho-syntactic Irony Markers:

This type of markers are based on the morphologic and syntactic levels of the utterance.

- Exclamation - Exclamation marks emphasize a sense of surprise on the literal evaluation that is reversed in the ironic reading (Burgers, 2010). Single or multiple exclamation marker are common in ironic utterances.

- Tag questions - Tag questions (e.g., “did n’t you?”, “are n’t we?”, “must we?”) are declarative or imperative statements that are turned into interrogatives. Authors of ironic utterances often explicitly state the irony and then attach a tag question to the statement.

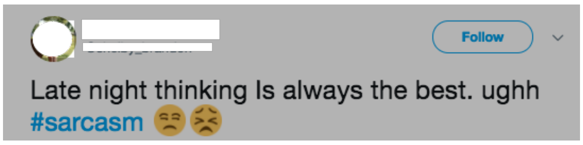
- Interjections - Interjections seem to undermine a literal evaluation and occur commonly in ironic utterances (e.g., “wow”, “yay”,“ouch” etc.). Similar to the tag questions interjections are often used in ironic utterances.

3.3.1.3 Typographic Irony Markers:

The last group of irony markers is the typographic markers.

- Capitalization - Users often capitalize words to represent their ironic use (i.e., use of “GREAT”, “SO”, and “WONDERFUL” in the ironic tweet “GREAT i’m SO happy shattered phone on this WONDERFUL day!!!”).

- Quotation mark - Users regularly put quotation mark to stress (i.e., “great” instead of “GREAT” in the above example) irony. Apart from stress, quotation marks highlight the non-standard meaning.



[redacted]

Late night thinking Is always the best. ughh #sarcasm 😒😣

Figure 3.1: Ironic utterance with emoji (best in color)

- Other punctuation marks - Punctuation marks such as “?”, “.”, “;” and their various uses (e.g., single/multiple/mix of two different punctuations) are used as features.

- Emoticon - Emoticons are fashionable to emphasize the ironic intent of the user. Such as in “I love the weather ;) #irony”, the emoticon “;)” (wink) alerts the reader to a possible ironic interpretation of weather (i.e., bad weather).

- Emoji - Emojis are like emoticons, but they are actual pictures instead of typographic and recently became very popular in social media since pictures represent sentiment better than typographic markers such as emoticons. Figure 3.1 shows a tweet with two emojis (i.e., “unassumed” and “confounded” faces respectively) used as markers.

3.3.2 Theoretical Overview of Irony Factors

Irony factors are inherent characteristics of the ironic utterance, that cannot be removed without destroying the irony (Attardo, 2000a). Identifying irony factors is complex since unlike the irony markers they are not a set of indicators. Burgers et al. (2012) compared various definitions of irony and found that irony should at least have five characteristics. They argue that every ironic utterance needs to have all of these five characteristics in order to be qualified as ironic. Therefore these characteristics are nothing but different types of irony factors. We briefly discuss them here.

- *Evaluativeness:* Irony factors are always evaluative. However, depending upon the use of terms (i.e., terms that could be substituted for its opposite term in the intended meaning) they are either *explicit* (i.e., explicitly evaluative) or *implicit* (i.e., implicitly evaluative). For instance, the utterance “That was a *great* investment idea!” is an example of an explicitly evaluative factor since *great* could be substituted for its opposite term to understand the intended meaning. Where as, the utterance “Investing in company X really earned me a lot of money!” is an example of an implicitly evaluative factor since the utterance does not directly discuss the merit of the investment idea. Rather, the utterance implies that the investment idea was not good. These two examples are from Burgers (2010).

- *Valence:* The reversal of valence is a well-discussed topic in pragmatics literature (Gibbs, 1986; Burgers, 2010). Irony can be *ironic praise* (i.e., irony with a positive literal meaning as in “Good game, Bob!”, when the game was poor, and *ironic blame* (i.e., irony with a negative literal meaning as in “Bad game, Bob!” when the game was great). We observe that the ironic praise is used more often than ironic blames in social media.[3] We model this characteristic of irony in the sense-disambiguation study (Section 3.5.2.1).

- *Target:* Irony and sarcasm are always aimed at somebody or something: its target (Burgers, 2010). Users often mock themselves or complain about their status or state of mind (standard in Twitter) or target other audiences (i.e., family, friends, etc.).

- *Relevance:* Irony should be relevant to the communicative situation. Giora (1995) stated that relevance of ironic utterances refers to the number of inferences that are needed to understand.

- *Incongruence:* Finally, irony is always dependent on some form of incongruence between the literal meaning of the utterance and the situation or context (Attardo, 2000a). It is possible that the incongruence is present in the utterance itself. For instance, in the running example, the incongruence between *love* and *going to the emergency room* is present in the utterance itself. This phenomenon often denoted as co-text incongruence. In contrary, the literal meaning could be incongruent with previous knowledge or utterances. For instance, UserB tweeted, “@UserA

[3] Burgers (2010); Jorgensen et al. (1984) also made similar observations.

Annotations	Source	Type	Experiments
self-label	$Twitter$	$Twitter_u$	irony markers
same	$Twitter$	$Twitter_u$	irony factors (reversal of valence, context incongruity)
same	$Twitter$	$Twitter_{u+c}$	role of conversation context
same	$Reddit$	$Reddit_u$	irony markers
same	$Reddit$	$Reddit_{u+c}$	role of conversation context
crowdsourced	IAC_{v2}	$IAC_{v2_{u+c}}$	role of conversation context

Table 3.1: Different corpus and associated experiments (u denotes only the utterances whereas $u + c$ denotes utterances with their context) for verbal irony identification

one more reason to feel really great #sarcasm" in reply to UserA's tweet "plane window shades are open ... so that people can see if there is fire". Here, UserB is ironic about the experience of flying. In that case, irony is incongruent in a particular type of context, conversation context. The details on using conversation context are in Section 3.6.

In this dissertation, we examine ironic utterances in isolation as well as with context (i.e., we study a particular type of context – conversation context in Section 3.6). At the same time we build our analysis on the irony factors and markers since these components can serve to differentiate between ironic and non-ironic utterances. In the following sections we present our empirical analysis of the irony markers and irony factors. We particularly analyze two irony factors — the reversal of valence property (Section 3.5.2.1) and semantic incongruence (Section 3.5.2.2 and Section 3.6). But first in the next section, we briefly revisit all the datasets that we have utilized in verbal irony identification.

3.4 Data and Preprocessing

In Chapter 2 we discuss all the datasets (i.e., self-labeled, such as tweets and $Reddit$ posts and crowdsource-labeled, such as the Internet Argument Corpus) used in the dissertation. Here is a quick recap of the data used in irony identification (Table 3.1).

Table 3.1 represents the different types of data and their description. Detail uses (i.e., quantity of training data used in a particular experiment) are described as part of subsequent sections when applicable.

3.4.1 Twitter Data

Our primary data source is Twitter, and we have relied upon the annotations that users assign to their tweets using hashtags. For instance, users label tweets with hashtags such as #sarcasm, #sarcastic, or #ironic. We used Twitter APIs to collect tweets for our research. As non-ironic utterances, we consider random objective utterances (i.e., tweets that do not have sentiment, irony, or irony related hashtags) as well as sentiment tweets. These sentiment tweets are composed of positive and negative sentiment. The positive tweets express direct positive sentiment (e.g., tweets with positive hashtags; #happy, #love, #lucky, etc.) and the negative tweets express direct negative sentiment (e.g., tweets with negative hashtags; #sadness, #angry, #frustrated etc.). See Table 2.2 for a full list of hashtags used. For brevity, we denote ironic utterances as I and non-ironic utterances as NI. The non-ironic random utterances are denoted as NI_{rand} and the sentiment (non-ironic) utterances are NI_{sent}.

3.4.2 Internet Argument Corpus

Oraby et al. (2016a) introduced the Sarcasm Corpus V2, a subset of the Internet Argument Corpus V2, which contain 9,400 posts labeled as sarcastic or non-sarcastic (balanced dataset). This corpus not only contains sarcastic posts but all the contexts (i.e., prior posts or prior turns) to which the sarcastic posts are replies to. Although the dataset described by Oraby et al. (2016a) consists of 9,400 post, only 50% of that corpus is currently available for research (4,692 altogether; balanced between sarcastic and non-sarcastic categories). Labels were obtained via crowdsourcing. This is the dataset we used in our study, particularly for the research on examining the role of conversation context (Section 3.6).

3.4.3 Reddit Corpus

Khodak et al. (2017) introduced the self-annotated Reddit Corpus, which is a very large collection of sarcastic and non sarcastic posts (over one million) from different subreddits. Similar to the IAC_{v2}, this corpus also contains the prior turn as conversation context (the prior turn is either the original post or a prior turn in the discussion thread that the current turn is a reply to). We have extracted 50K reddit posts and their context for investigating the role of irony markers as well as research on conversation context. As shown in Table 3.1, this corpus is self-labeled.

3.4.4 Dense Representation of Input Data

Beside various discrete features such as lexical and pragmatic features, we have heavily used dense representation of words in our research. Particularly, we have used word-embeddings for modeling the semantics of words (vector based representation). In this section we propose a brief introduction of the various types of word embeddings employed in this dissertation.

Recently, a new set of language techniques in natural language processing (NLP) have received a major success where words are mapped to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with much lower dimension. These techniques involve learning representation of words from its context where the models typically train on a very large corpus (i.e., Wikipedia, common crawl corpora of billion web pages etc.). Methods to generate this mapping include neural networks and dimensionality reduction on the word co-occurrence matrix (Pennington et al., 2014; Mikolov et al., 2013a). The vector representations in word embeddings show excellent mathematical relations: for instance the *geometric distance* between the vectors between *Germany* and *Berlin* is similar to the *distance* between the vectors between *France* and *Paris* and so on. Word embeddings, when used as the underlying input representation (i.e., vectors for supervised learning such as SVM linear kernel or input for neural network models) have been shown to boost the performance in NLP tasks such as syntactic parsing, sentiment analysis, information extraction tasks (i.e., relation extraction) and machine translations (Socher et al., 2010; Zeng et al., 2014; Zou et al., 2013; Dos Santos and Gatti, 2014). Below are short description of the word embedding approaches.

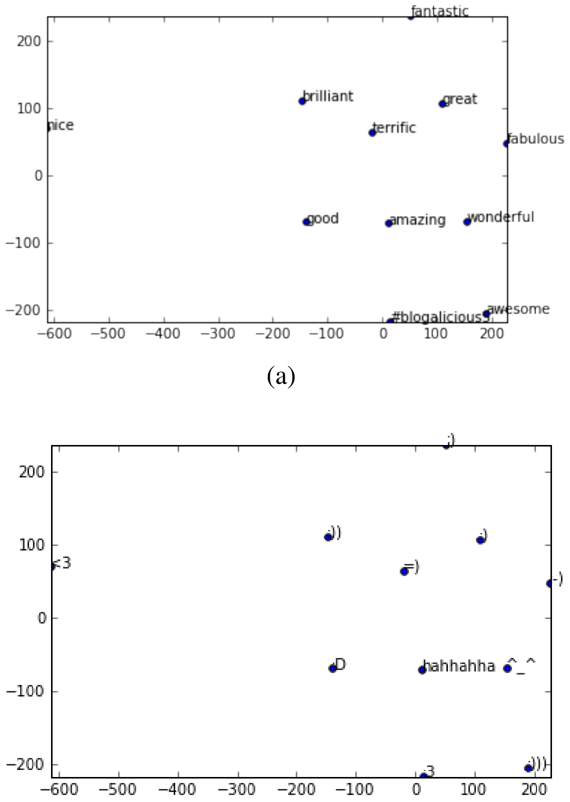
- *Weighted Textual Matrix Factorization (WTMF):* Low-dimensional vectors have been used in Word Sense Disambiguation (WSD) tasks, since they are computationally efficient and provide better generalization than surface words. A dimension reduction method is Weighted Textual Matrix Factorization (WTMF), which is designed specifically for short texts, and has been successfully applied in WSD tasks (Guo and Diab, 2012a). WTMF typically models unobserved words (i.e, not seen during training), thus providing more robust embeddings for short texts such as tweets.

- *word2vec Representation:* We use both the Skip-gram model and the Continuous Bag-of-Words (CBOW) model (Mikolov et al., 2013a,b) as implemented in the word2vec gensim python library.[4] Given a window size of n words around a word w, the skip-gram model predicts the neighboring words given the current word. In contrast, the CBOW model predicts the current word w, given the neighboring words in the window. For research that are based on tweets, we build our own word2vec models where we considered a context window of ten words. For research based on discussion forums, we use the standard Google n-gram word embedding.

- *GloVe Representation:* GloVe (Pennington et al., 2014) is a word embedding model that is based upon weighted least-square model trained on global word-word co-occurrence counts instead of the local context used by word2vec. We retrain GloVe model with the same number of tweets that we utilized for Word2Vec. For WTMF, Word2Vec, and GloVe, vector dimension was set to 100.

We build the above word embedding models in an unsupervised fashion with around 3 millions tweets from our Twitter collection. In each of the three models, each word w is represented by its d-dimensional vector $\vec{w}$ of real numbers, where d=100 for all of the embedding algorithms in our experiments. We use the context window of ten words to build the embeddings. In all the subsequent research using Twitter data, we are using this particular word embeddings. However, for research using discussion forum data (e.g., $Reddit$ and IAC_{v2}) we use the standard Google n-gram off-the-shelf word

[4] https://radimrehurek.com/gensim/models/word2vec.html



(a)

(b)

Figure 3.2: Examples of word embeddings via t-SNE ((a): “great”; (b) “:)”

embeddings (Mikolov et al., 2013a).

3.5 Utterance-level Analysis

In this section, we discuss our research on verbal irony identification while treating the utterance in isolation. We first describe our research on examining the role of irony markers.

3.5.1 Empirical Analysis of Irony Markers

In Section 3.3.1 we introduced the theoretical framework of irony markers and discussed different types of markers, such as the tropes, morpho-syntactic, and typographic markers. In this section, we examine the role of irony markers, particularly in the context of utterance level classification. We address the following research questions:

- RQ1: Are irony markers discriminative features for automatic irony identification?

- RQ2: Do irony markers differ across platforms such as Twitter and discussion forums ($Reddit$)?

- RQ3: Can irony markers generalize well over data collected from different time periods?

For the first question, we want to analyze whether the markers are sufficient as discriminative features to classify ironic utterances vs. non-ironic utterances with high accuracy.

For the second question, we compare irony markers across two platforms, Twitter and $Reddit$. We also investigate different subreddits (e.g., technology vs. political forums) to identify whether there are specific markers used more frequently in different subreddits.

The third question arises from a particular type of sampling bias in social media platform such as Twitter. An important issue while working with social media is that users' sentiments, opinions or beliefs might be about specific events that are temporary and thus systems trained on social media content (e.g., tweets) collected from a period of time might not generalize well when tested on tweets from a different period of time. Tufekci (2014) argued that language in Twitter is continuously evolving and users are bringing new terms, hashtags and other linguistic expressions to present their sentiment and opinions. Thus, we need to examine whether the same irony markers are consistently used across different topics, events, and time periods. In this scenario, it is unclear whether specific irony markers are related to specific trending events or are consistently used by users to express various types of verbal irony in social media. This question is addressed in Section 3.5.1.2.

To answer the first and second question we conduct a series of classification experiments with features that are based on irony markers. As stated earlier, irony category is denoted as I and the non-ironic categories are denoted respectively as NI_{rand} and NI_{sent} (details in Section 3.4).

3.5.1.1 Irony Markers as Discriminative Features

We first address RQ1 and RQ2. Below we describe the irony markers we have used as features to identify ironic utterances from $Twitter$ and $Reddit$. These markers were already introduced in Section 3.3.1. Here we discuss the implementation details.

Tropes: Tropes are a type of figurative language.

- Metaphors - We have drawn metaphors from two different sources (e.g., 884 and 8,600 adjective/noun metaphors from Tsvetkov et al. (2014) and Gutiérrez et al. (2016), respectively) and used them as binary features. Here, all the metaphors are bigrams (e.g., 'black hole")

  We also evaluate the metaphor detector described in Rei et al. (2017) over $Twitter$ and $Reddit$ datasets to examine whether the metaphor detector finds new metaphors. We consider metaphor candidates that have precision $\geq$ 0.75; (see (Rei et al., 2017)) and use them as binary features. Note, similar to the metaphor sources, the metaphor detector also only works on metaphors of two words and does not detect metaphor such as "all the world's a stage". We consider this is a limitation of metaphor marker.

- Hyperbole - We use terms that are denoted as "strong subjective" (positive/negative) in the Multi-Perspective Question Answering (MPQA) corpus (Wilson et al., 2005) as hyperboles or intensifiers. Apart from using hyperboles directly as binary features we also use their sentiment (i.e., positive/negative) as features.

- Rhetorical Questions - We follow the hypothesis from Oraby et al. (2017) who found RQs by searching questions in the middle of an utterance since question followed by text cannot be typical information seeking question. The presence of RQ is used as a binary feature.

Morpho-syntactic Irony Markers: This type of markers works on the morphological levels of the utterance.

- Exclamation - We use two binary features, single and multiple uses of the marker.

- Tag questions - We drew a list of tag questions from grammar site and use them as binary indicators.[5]

- Interjections - Similar to tag questions we assembled interjections (e.g., “wow”, “yay”,“ouch” etc., a total of 250) from different grammar sites and employ them as features.

Typographic Irony Markers:

- Capitalization - Binary feature to check whether all the characters of a word are uppercase.

- Quotation marks - Users regularly put quotation mark to stress irony (binary feature).

- Other punctuation marks - Punctuation marks such as “?”, “.”, “;” and their various uses (e.g., single/multiple/mix of two different punctuations) are used as features.

- Emoticon - We collected (a) a comprehensive list of emoticons (over one-hundred) from Wikipedia and (b) use standard regular expressions to identify emoticons.[6] Aside from using the emoticons directly as binary features, we use their sentiment as well (e.g., “wink” is regarded as positive sentiment in the MPQA corpus) as features.

- Emoji - Emojis are like emoticons, but they are actual pictures instead of typographic and recently became very popular in social media since pictures represent sentiment better than typographic. We utilize an emoji library of 1,400 emojis to identify the particular emoji appear in ironic utterance and treat them as binary features.[7]

[5] http://www.perfect-english-grammar.com/tag-questions.html

[6] http://sentiment.christopherpotts.net/code-data/

[7] https://github.com/vdurmont/emoji-java

Features	Category	P	R	F1
all	I	**78.16**	**75.18**	**76.64**
	NI_{rand}	**76.10**	**78.98**	**77.51**
all - tropes	I	**78.98**	44.33	56.79
	NI_{rand}	61.30	88.20	72.33
all - morpho_syntactic	I	60.63	73.26	66.35
	NI_{rand}	66.22	52.42	58.52
all - typography	I	75.16	72.55	73.83
	NI_{rand}	73.47	76.02	74.73

Table 3.2: Ablation tests of irony markers (I vs. NI_{rand}). **bold** are best scores (in %).

We first conduct a classification task to decide whether an utterance (e.g., a tweet or a $Reddit$ post) is ironic or non-ironic, exclusively based on the irony marker features. We adopt a binary Support Vector Machines (SVM) classification setup (linear kernel from Fan et al. (2008)) with category weights inversely proportional to utterance frequencies. Linear kernel is preferred since this kernel provides us the opportunity to directly look at the weights of the features that informs us about the discriminating power of the features. For instance if the feature "presence of an emoticon x" has high weight for the class I we interpret that this emoticon x feature is a powerful discriminatory feature to identify verbal irony.

Table 3.2 shows the accuracy of identification of ironic (I) instances vs. random (NI_{rand}). Likewise, Table 3.3 shows the accuracy of identification of ironic (I) instances vs. sentiment (NI_{sent}) utterances via ablation tests for tweets. For both experiments, we used 300K training data and 40K test data (balanced training data between the two categories). We observe that the F1 score with all features for the I class in Table 3.2 is higher than Table 3.3. This is expected since NI_{sent} also use many markers such as hyperbole or emoticons and thus NI_{sent} is more similar to I than NI_{rand}. Both tables also show that removal of the tropes (e.g., mainly the hyperboles) have the maximum effect on the F1 scores for the I category since the recall drops in both cases. Finally, we also observe that the removal of morpho-syntactic markers (e.g., exclamations, interjections) have more effect on the I vs. NI_{rand} experiment than on the I vs. NI_{sent}.

Table 3.4 presents the result of I vs. NI classification using irony markers as features for $Reddit$ posts. We use the full $Reddit$ corpus (50K posts) for the experiment.

Features	Category	P	R	$F1$
all	I	66.93	**77.32**	**71.75**
	NI_{sent}	**73.13**	61.78	**66.97**
all - tropes	I	**67.70**	48.00	56.18
	NI_{sent}	59.70	**77.09**	**67.29**
all - morpho_syntactic	I	63.59	**78.09**	70.10
	NI_{sent}	**71.59**	55.27	62.38
all - typography	I	57.30	77.95	66.05
	NI_{sent}	65.49	41.86	51.07

Table 3.3: Ablation tests of irony markers (I vs. NI_{sent}). **bold** are best scores (in %).

Features	Category	P	R	$F1$
all	I	**73.16**	48.52	**58.35**
	NI	**61.49**	**82.20**	**70.35**
all - tropes	I	71.45	**50.36**	**59.08**
	NI	**61.67**	79.88	**69.61**
all - morpho_syntactic	I	58.37	**49.36**	53.49
	NI	56.13	64.8	60.16
all - typography	I	73.29	48.52	58.39
	NI	**61.52**	**82.32**	**70.42**

Table 3.4: Ablation Tests of irony markers for $Reddit$ posts. **bold** are best scores (in %).

In comparison to the results on $Twitter$ (Table 3.2 and Table 3.3), Table 3.4 shows that removal of typography such as emoticons do not affect the F1 scores (in fact, the F1 increases a little with removal of typography markers) whereas morpho-syntactic markers, e.g., tag questions, interjections have more effect on the F1 for $Reddit$ posts. This is expected since the use of typography markers such as emojis and emoticons are less frequent in $Reddit$ than $Twitter$.

Table 3.5 and Table 3.6 represent the top features for both categories based on the feature weights learned during the SVM training for $Twitter$ (from both I vs. NI_{rand} and I vs. NI_{sent} experiments) and $Reddit$, respectively. Table 3.5 shows typographic features such as emojis and emoticons have the highest feature weights for both categories ($Twitter$). Interestingly, we observe for ironic tweets users often express negative sentiment directly via emojis (e.g., angry face, rage) whereas for non-ironic utterances, emojis with positive sentiments (e.g., hearts, wedding) are familiar. For the Irony category, interjections emphasize strong positive emotions (i.e., “yay”, “yippee”, “aww”). As users are not able to use any spoken communication type features (i.e.,

Category	Top features
I	emoticons: annoyed ("-_-"), perplexed (":-/"); emojis: angry face/monster, unamused, expressionless, confounded, rage, neutral_face, thumbsdown; negative_tag questions ("is n't it?", "don't they?"), interjections (e.g., "yay", "yippee", "aww"), multiple_exclamations ("!!!")
NI	emojis: birthday, tophat, hearts, wedding, rose, ballot_box_with_check; quotations, hashtags (positive sentiment), emoticons: happy (":)"), overjoyed ("∧_∧")

Table 3.5: Irony markers based on feature weights for $Twitter$

Category	Top features
I	exclamation (single, multiple), negative_tag questions ("is n't it?", "don't they?"), interjections, presence of metaphors, positive sentiment hyperbolic words (e.g., "notably", "goodwill", "recommendation")
NI	negative sentiment hyperbolic words (e.g., "vile", "lowly", "fanatic"), emoticon: laugh (":))"), positive_taq questions ("is it?", "are they?"), punctuations such as periods/multiple periods

Table 3.6: Irony markers based on feature weights for $Reddit$

intonation, nasalization, excessive stress to represent strong sentiment), they tend to use interjections to convey verbal irony. Similar to interjections, it is common to apply multiple exclamation marks (to represent exaggerated stress) in ironic utterances, many times multiple ("!!!") ones. From Table 3.6 for $Reddit$, we observe instead of emojis, other markers such as exclamation marks, negative tag questions, metaphors are discriminatory markers for irony category. In contrary, for the non-irony category, positive tag questions and negative sentiment hyperboles are influential features.

We also conduct frequency analysis of the irony markers on both platforms. Table 3.7 shows the frequency analysis of the markers on Twitter and $Reddit$. Likewise, we conduct frequency analysis of markers on different subreddits from $Reddit$ corpus.

We report the mean of occurrence per utterance and the standard deviation (SD) of each irony marker. Table 3.7 demonstrates that markers such as hyperbole, punctuations, and interjections are popular in both platforms. Emojis and emoticons, although the two most popular markers in $Twitter$, are almost unused in $Reddit$. In contrast,

Irony Markers		Corpus	
Type	Marker	*Twitter*	*Reddit*
Trope	Metaphor	0.02 (0.16)	0.01 (0.08)
	Hyperbole	0.45 (0.50)	0.43 (0.50)
	RQ	0.01 (0.08)	0.15 (0.36)
Morpho_Syntactic	Exclamation	0.02 (0.16)	0.19 (0.39)
	Tag Question	0.02 (0.10)	0.08 (0.26)
	Interjection	0.22 (0.42)	0.32 (0.46)
Typographic	Capitalization	0.03 (0.16)	0.10 (0.30)
	Quotation	0.01 (0.01)	-
	Punctuations	0.10 (0.29)	0.47 (0.50)
	Emoticon	0.03 (0.14)	0.001 (0.03)
	Emoji	0.05 (0.22)	-

Table 3.7: Frequency of irony markers in ironic utterances from $Twitter$ and $Reddit$ platforms. The mean and the Standard Deviation (SD, in bracket) are reported.

exclamations and RQs are more common in the $Reddit$ corpus. We also combine each marker to the group they belong to (i.e., either of the trope, morpho-syntactic and typographic) and compare the means between a pair of the groups via independent t-tests. We found that the difference of means is significant ($p \leq 0.005$) for all pair of groups across the two platforms.

Since $Reddit$ posts are collected from different topics (i.e., subreddits), it is important to study whether particular markers are more common to a particular subreddit. Thus, we conduct an experiment over irony posts from specific subreddits. We look at political subreddits (e.g., hillary, the_donald), forums related to sports (e.g., nba, football, soccer), religion subreddits, and technology subreddits. Table 3.8 presents the mean and SD for each subreddit genre. We observe that users use tropes such as hyperbole and rhetorical questions, morpho-syntactic markers such as exclamation, and interjections, and multiple-punctuation marks more in politics and religion than in technology and sports subreddits. This is expected since subreddits regarding politics and religion are often more controversial than technology and sports.

3.5.1.2 Generalization of Irony Markers

To address the third research question RQ3 we are interested in evaluating the generalization power of verbal irony detection systems when trained and tested on data sets that span different time frames. We want to examine whether the markers are characteristics

Irony Markers		Genres			
Type	Marker	Technology (a)	Sports (b)	Politics (c)	Religion (d)
Trope	Metaphor	0.01 (0.06)	0.002 (0.05)	0.02 (0.12)	0.01 (0.10)
	Hyperbole	0.19 (0.39)	0.34 (0.48)$^{a^{**}}$	0.74 (0.44)$^{(a,b)^{**}}$	0.76 (0.43)$^{(a,b)^{**},c^{*}}$
	RQ	0.06 (0.23)	0.11 (0.32)$^{a^{**}}$	0.22 (0.41)$^{(a,b)^{**}}$	0.2 (0.4)$^{(a,b)^{**}}$
MS	Exclamation	0.09 (0.29)	0.14 (0.34)$^{a^{**}}$	0.42 (0.49)$^{(a,b)^{**}}$	0.37 (0.48)$^{(a,b,c)^{**}}$
	Tag Question	0.03 (0.16)	0.05 (0.23)$^{a^{**}}$	0.11 (0.32)$^{(a,b)^{**}}$	0.1 (0.30)$^{(a,b)^{**}}$
	Interjection	0.13 (0.34)	0.23 (0.42)$^{a^{**}}$	0.45 (0.50)$^{(a,b)^{**}}$	0.52 (0.5)$^{(a,b,c)^{**}}$
Typographic	Capitalization	0.04 (0.19)	0.08 (0.27)$^{a^{**}}$	0.20 (0.40)$^{(a,b)^{**}}$	0.1 (0.31)$^{(a,b,c)^{**}}$
	Punctuations	0.23 (0.42)	0.45 (0.50)$^{a^{**}}$	0.84 (0.36)$^{(a,b)^{**}}$	0.89 (0.31)$^{(a,b,c)^{**}}$

Table 3.8: Frequency of irony markers in different genres (subreddits). For brevity, Morpho_Syntactic is represented as MS. The mean and the Standard Deviation (SD, in bracket) are reported.$^{x^{**}}$ and $^{x^{*}}$ depict significance at $p \leq 0.005$ and $p \leq 0.05$, respectively.

of trending events or are generalizable over social media data collected over extended time – we used the time-stamp of the tweets and split the dataset by weeks, training on each week and testing on all the other weeks.[8] We have tweets collected over seventeen weeks between 2013-2014 (for brevity, $S1314$ corpus) and over five weeks between 2014-2015 (for brevity, $S1415$ corpus).

We created balanced training datasets for each of the seventeen weeks in $S1314$. Each training set consists of 24,000 instances (balanced between I and NI_{sent} categories) and corresponding test sets for each weekly time frame consisting of 6,000 instances (again, balanced between I and NI_{sent} categories). For the $S1415$ corpus, we noticed that the Twitter-API did not retrieve as many sarcastic tweets (per week) as in the previous year, thus the number of training and test instances per week was 20,800 and 5,200 respectively balanced between the I and NI_{sent} categories. However, the order of magnitude seems to be consistent in the two corpora representing two different

[8] trending events are those events that are popular for a short period.

years. Since the non-sarcastic category consists of tweets collected using various hashtags, we kept the actual distribution of these hashtags when building the training and test sets for each week time frame. In other words, we use the actual distribution of the hashtags (e.g., “love”, “mature” etc.) of the tweets. Given the $S1314$ corpus span over 17 weeks of tweets, thus we conducted 17 * 17 = 289 experiments (training and test on each week in $S1314$).

We implement a simple heuristic to chose the most discriminative features. First, we obtain the weight vector $w \in \mathbb{R}$ from the linear SVM model to decide the relevance of the features. Next, we rank the features according to the w_j value for the jth feature in each week (training data) from $S1314$ and $S1415$, respectively. Irony markers that appeared in at least two highest positive (indicative of the ironic category I) or negative (indicative of the non-ironic NI_{sent} category) weight lists across all weeks are considered for our analysis. We observe there are several consistent features with high feature weights occurring in every week and most of the features are verbal irony markers. The most discriminative ones are listed below.

- interjections emphasizing strong positive emotions (e.g., “whoohoo”, “hooray”)
- hyperbolic markers that denote the extreme end of a normative scale (e.g., “brilliant”, “exciting”, “genius”)
- uppercase words that serve as proxy for accented speech (e.g., “GREAT”, “NEVER”)
- hashtags such as “#funny”, “#humor”
- multiword expression given as hashtags (e.g., “#soexcited”, “#notreally”)
- use of quotation, emoticons such as winking- face (“;)”) and sticking tongue out (“:P”)
- alternate spelling of words to attract readers (e.g., ‘yeah” as “yay”, “yayy”, “yayyy”)
- intensifiers, interjections, punctuations (multiple question marks and exclamations)

In other words, irony markers can very well generalize over different time periods. We discover that no matter when the utterance appeared between 2013-2014 or 2014-2015, markers in the above list are always associated strongly with ironic utterances. One limitation of this method is that the time period is relatively short. So we can hypothesize that there could be more variations over longer time frames.

So far to address the research questions related to the irony markers we have shown that (1) certain irony markers are useful predictive features for verbal irony, (2) we can identify particular irony markers that are associated with specific social media platform (e.g., $Twitter$ or different subreddits from $Reddit$), and (3) they occur consistently over different time-frames.

However, the results in Table 3.2 and Table 3.3 show that many ironic utterances do not contain any indicators. In other words, the ironic nature of these utterances is marked by other characteristics, possibly by irony factors (Attardo, 2000b).

3.5.2 Modeling of Irony Factors

In this dissertation, we model two types of irony factors, the reversal of valence and the effect of meaning incongruence in ironic utterances. Reversal of valence states that the intended meaning of the ironic statement is opposite to its literal meaning. We observe in numerous utterances in Twitter, users user *sentiment* words in conjunction with negative situations to express the verbal irony. For instance, in our running example, the author used the positive sentiment word “love” to express the verbal irony in conjunction with the negative situation “going to the emergency room”. Thus, to identify the reversal of valence, it is crucial to recognize the sense (i.e., literal or ironic) of the sentiment word in an utterance. To solve this problem we reframed the task of detecting verbal irony detection to a sense-disambiguation problem: given an utterance that contains a sentiment word we identify its sense (i.e., literal vs. ironic). We refer to this task as the Literal Ironic Sense Disambiguation (LISD) task.

3.5.2.1 Literal/Ironic Sense Disambiguation (LISD) task

The research question we address in this section is,

- RQ4: How to detect whether a word is used in a literal or ironic sense in an utterance?

We address two challenges here:

- how to collect a set of words (hereafter, *target words*) that can have either literal or ironic meaning depending on context. For instance, “love” from the running example “ love going to ... ” is a target word.
- given an unknown utterance and a target word, how to automatically detect whether the target word is used in the literal or the ironic sense.

3.5.2.1.1 Identifying Target Words: We conducted a crowd-sourcing experiment to identify the target words. The task is framed as follows: given a ironic message (IM), Turkers on Amazon Mechanical Turk (MTurk) are asked to re-write the message so that the new message is likely to express the speaker's intended meaning (H^{i}_{int}). We collected 1,000 IMs from Twitter using the Twitter APIs and received 5,000 H^{i}_{int}s from the Turkers.[9] Examples of an original ironic message (1) and three messages generated by the Turkers (a,b,c) are below:

1. IM: I am so $\underline{\text{happy}}$ that I am going back to the emergency room.
   (a) H^{1}_{int}: I $\underline{\text{don't like}}$ that I have to go to the emergency room again.
   (b) H^{2}_{int}: I am so $\underline{\text{upset}}$ I have to return to the emergency room.
   (c) H^{3}_{int}: I'm so $\underline{\text{unhappy}}$ that I am going back to the emergency room.

From the above examples, we see that aligning the ironic message (IM) to the author's intended meanings (H^{i}_{int}) will allow us to detect that “happy” can be aligned to “don't like”, “upset”, and “unhappy”. Based on this alignment, “happy” will be considered as a target word for the LISD task. We treat the IM-H^{i}_{int} data as a parallel corpus for alignment (Statistical Machine Translation parlance) and apply the monolingual alignment algorithm as developed by Barzilay and McKeown (2001).

[9] We explain the crowd-sourcing experiment in detail in Chapter 4 since we utilize the same datasets.

This algorithm is used for two specific reasons. First, our dataset is similar in nature to the parallel monolingual dataset used in Barzilay and McKeown (2001), and thus lexical and contextual information from tweets can be used to extract the candidate targets words for LISD. For instance, we can align the [IM] and [H^3_{int}] (from the above examples), where except for the words happy and unhappy, the majority of the words in the two messages are anchor words (i.e., specific words that are common in both the messages) and thus happy and unhappy can be extracted as paraphrases via co-training. We used Tweet NLP to extract parts-of-speech of the words in the utterances (Gimpel et al., 2011). Second, Bannard and Callison-Burch (2005) noticed that the co-training method proposed by Barzilay and McKeown (2001) requires identical bounding substrings and has a bias towards single words while extracting paraphrases. This apparent limitation, however, is advantageous to us because we are specifically interested in extracting target unigrams. Co-training extracted 367 extracted pairs of paraphrases.

Next we considered a statistical machine translation (SMT) alignment method - IBM Model 4 with HMM alignment implemented in Giza++ (Och and Ney, 2000). We used Moses software (Koehn et al., 2007a) to extract lexical translations by aligning the dataset of 5,000 SM-IM pairs. From the set of 367 extracted paraphrases using (Barzilay and McKeown, 2001)'s approach, we selected only those paraphrases where the lexical translation scores ϕ (resulted after running Moses) are $\geq$ 0.8. After filtering via translation scores and manual inspection, we obtained a set of 80 semantically opposite paraphrases. Given this set of semantically opposite words, the words that appear in the ironic messages were consider our target words for LISD (70 target words after lemmatization). They include verbs, such as "love" and "like", adjectives, such as "brilliant", "genius", and adverbs, such as "really".[10]

3.5.2.1.2 Disambiguation Task: Once the MTurk task and alignment algorithms have finished collecting target words we propose several distributional semantics methods to classify ironic vs. literal meaning of words. We consider two classification tasks:

[10] In the next chaper we delve into understanding the rephrasing strategies of the Turkers and subsequently we describe the MTurk experiments in detail.

Target words	Sense	Utterance
great	I	. . . starting off the new year great !!!!! sick in bed . . .
	L	. . . you don't need a record label to have great music . . .
	L_{sent}	. . . i'm in love with this song great job justin . . .
proud	I	yay something to be proud of 3rd poorest in the NATION . . .
	L	im filipino with dark brown eye and forever true and proud . . .
	L_{sent}	but i'm proud of all the beliebers AROUND THE WORLD . . .

Table 3.9: Examples of target words(underlined) and their senses

love(26802), like(14995), great(14495), good(11624), really(9825), right(6771), fun(6603), best(6182), better(5960), glad(5748), yeah(5504), nice(4443), awesome(4196), excited(4027), always(3807), happy(3098), cool(2705), amazing(1952), favorite(1883), perfect(1792), wonderful(1749), wonder(1476), lovely(1424), super(1390), fantastic(1369), joy(1176), cute(1007), beautiful(981), sweet(800), hot(729), proud(703), shocked(645), interested(624), brilliant(576), genius(481), attractive(449), mature(427)

Table 3.10: Target words and # of training instances per class

verbal irony vs. literal (for brevity, I vs. L) and verbal irony vs. sentiment (for brevity, I vs. L_{sent}), and aim to collect a balanced data set for each target word.

The tweets that contain a target word and are annotated with the #sarcasm, #sarcastic, and #irony hashtags represent the *ironic sense* (I). In contrary, tweets that contain the target word and are *not* annotated with the #sarcastic or #sarcasm hashtags designate the *literal sense*. Table 3.9 shows examples of two target words ("great" and "proud") and their ironic sense (I) and literal sense (L). Also, for the *literal sense*, we consider a special case, where the tweets are labeled with either positive or negative hashtag (e.g., #happy, #sad) (Table 3.9). As before, we represent these tweets as L_{sent}. We used a setup where 80% of data is used for training, 10% for development, and 10% for the 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 LISD experiments. Table 3.10 shows all the target words and their corresponding number of *training* instances for each sense (I and L/L_{sent}). The size of training data ranges from 54 K for the target word "love" to 900 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, especially for the methods that use word embeddings.[11]

Computational Models and Experimental Setup: We consider two strategies for the LISD tasks: (1) in the distributional approach, context vectors derived from the training data represent each sense of a target word and (2) classification approach (I vs. L; I vs. L_{sent}) for each target word.

(1) Distributional Approaches: The Distributional Hypothesis is derived from the semantic theory of language use, i.e., words that are used and occur in the same contexts tend to purport similar meanings (Harris, 1954). Distributional semantic models (DSMs) use vectors that represent the contexts (e.g., co-occurring words) in which candidate words appear in a corpus, as proxies for meaning representations. Geometric techniques such as cosine similarity are then applied to these vectors to measure the similarity in meaning of corresponding words.

The DSMs are a natural approach to model our LISD task. For each target word t we build two context-vectors that represent the two senses of the target word t using the training data: one for the *ironic* sense I using the training data for t ($\vec{v_i}$) and one for the *literal* sense L using the literal sense training data for t ($\vec{v_l}$).[12] Given a test utterance u containing a target word t, we first represent the target word as a vector $\vec{v_u}$ using all the context words inside u. To predict whether t is used in a literal or ironic sense in the test message u we simply apply geometric techniques (e.g., cosine similarity) between $\vec{v_u}$ and the two sense vectors $\vec{v_i}$ and $\vec{v_l}$, choosing the one with the maximum score.

We created the sense vectors in two ways.

- Using positive pointwise mutual information model (PPMI) (Church and Hanks, 1990): Based on t's context words c_k in a window of ten words, we separately computed PPMI for ironic and non-ironic senses using t's training data. The size of the context window used in DSMs is generally between two and five, and in our experiments we used a window of ten words since tweets often include meaningful words/tokens at the end of the tweets (e.g., interjections, such as "yay",

[11] The datasets as well as the embeddings used in the experiments are available at https://github.com/debanjanghosh/sarcasm_wsd.

[12] In the remaining of this section we will only mention L and not L_{sent} for clarity and brevity.

"ohh"; upper-case words, such as "GREAT"; novel hashtags, such as "#notreally", "#lolol"; emoticons, such as ":("). We sorted the context words based on the PPMI scores and for each target word t we selected a maximum of 1,000 context words per sense to approximate the two senses of the target word (i.e., the vectors $\vec{v_i}$ and $\vec{v_l}$ for each target word t consist of a maximum of 1,000 words). Table 3.11 shows some target words and their corresponding context words that were selected based on high PPMI scores. To predict whether t is used in a ironic or non-ironic sense in the test message u we simply apply the cosine similarity to the $\vec{v_u}$ (vector representation of the target word t in the test message u) and the two sense vectors $\vec{v_i}$ and $\vec{v_l}$ of t, choosing the one with the maximum score. All vector elements are given by the tf-idf values of the corresponding words. This approach, denoted as the "PPMI baseline", is the baseline for our DSM experiments.

- using word embedding model: We enhance the representation of context vectors to represent each word in the context vector by its word embedding. We experiment with three different methods of obtaining word embeddings: Weighted Textual Matrix Factorization (WTMF) (Guo and Diab, 2012b); *word2vec* that implements the skip-gram and continuous bag-of-words models (CBOW) of (Mikolov et al., 2013a), and GloVe (Pennington et al., 2014). Details of the word embedding models are in Section 3.4. Here the difference from the baseline is twofold: First, all vectors elements are word embeddings (i.e., 100-d vectors). Second, we use the *maximum-valued matrix-element* (MVME) algorithm introduced by Islam and Inkpen (2008), which has been shown to be particularly useful for computing the similarity of short texts. Once the similarity is computed via MVME algorithm (details are in Algorithm 1), similar to the PPMI approach we use a similarity measure between the $\vec{v_u}$ and the two sense vectors $\vec{v_i}$ and $\vec{v_l}$ of t, choosing the one with the maximum score.

We modify this algorithm to use word embeddings ($MVME_{we}$). The idea behind the MVME algorithm is that it finds a one-to-one "word alignment" between two utterances based on the pairwise word similarity. Only the aligned words contribute to

Target	Senses	Context Vector
love	I	ignored, being, waking, work, sick, #not
	L	please, follow, ♡, her, :)
	L_{sent}	happy, family, blessed, cute, birthday
fun	I	work, tomorrow, homework, friday, sleep
	L	hope, join, girl, game, friend
	L_{sent}	#friends, #family, weekend, amazing, #christmas
joy	I	working, snow, waking, studying, sick
	L	yesterday, sweet, special, prayer, laughter
	L_{sent}	wishing, warmth, love, christmas, peace

Table 3.11: Target words and their context words

Algorithm 1 Description of $MVME_{we}$ Algorithm

```
procedure $MVME_{we}(v_s,v_u)$
    $v_{s_{words}} \leftarrow v_s.elements()$
    $v_{u_{words}} \leftarrow v_u.elements()$
    $M[v_{s_{words}}.size(), v_{u_{words}}.size()] \leftarrow 0$
    for $k \leftarrow 0, v_{s_{words}}.size()$ do
        $c_k \leftarrow v_{s_{words}}[k]$
        $\vec{c_k} \leftarrow getEmbedding(c_k)$
        for $j \leftarrow 0, v_{u_{words}}.size()$ do
            $w_j \leftarrow v_{u_{words}}[j]$
            $\vec{w_j} \leftarrow getEmbedding(w_j)$
            $M[k][j] \leftarrow cosine(\vec{c_k}, \vec{w_j})$
        end for
    end for
    while True do
        repeat
            $max \leftarrow getMax(M)$
            $Sim \leftarrow Sim + max$
            $r_m, c_m \leftarrow getRowCol(M, max)$
                                        ▷ Remove $r_m$ row and $c_m$ column from M
            $remove(M, r_m, c_m)$
        until $max > 0$ Or $M.size() > 0$
    end while
    Return $Sim$
end procedure

procedure GETEMBEDDING(word)
    Return $we_{model}[word]$
end procedure
procedure GETROWCOL(M,max)
    $row, col \leftarrow M.indexOf(max)$
    Return $row, col$
end procedure
```

the overall similarity score. Algorithm 1 presents the pseudocode of our modified algorithm for word embeddings, $MVME_{we}$. Let the total similarity between $\vec{v_s}$ and $\vec{v_u}$ be Sim. For each context word c_k from $\vec{v_s}$ and each word w_j from $\vec{v_u}$, we compute a matrix where the value of the matrix element M_{jk} denotes the cosine similarity between the embedded vectors $\vec{c_k}$ and $\vec{w_j}$ [lines 5 -13]. Next, we first select the matrix cell that has the highest similarity value in M (max) and add this to the Sim score [lines 16-17]. Let the r_m and c_m be the row and the column of the cell containing max (maximum-valued matrix element), respectively. Next, we remove all the matrix elements of the r_m-th row and the c_m-th column from M [line 20]. We repeat this procedure until we have traversed through all the rows and columns of M or max = 0 [line 21].

(2) Classification Approaches: The second approach for our LISD task is to treat it as a binary classification task to identify the ironic or literal sense of a target word t. We have two classification tasks: I vs. L and I vs. L_{sent} for each of the 37 target words. We use the libSVM toolkit (Chang and Lin, 2011). Development data is used for tuning parameters. We propose two classification setups: (1) SVM classification (SVM_{bl}) using lexical features such as n-grams (i.e., unigrams and bigrams), LIWC dictionary to identify the pragmatic features Pennebaker et al. (2001b), interjections (e.g., “ah”, “oh”, “yeah”), punctuations, exclamations (e.g., “!”, “?”), and emoticons for the SVM baseline. (2) A new kernel $kernel_{we}$ to compute the semantic similarity between two tweets u_r and u_s using the $MVME_{we}$ method introduced for the DSM approach, and the three types of word embeddings (WTMF, word2vec, and GloVe). The similarity measure in the kernel is similar to the algorithm $MVME_{we}$ described in Algorithm 1, but instead of measuring the similarity between the sense vectors of t ($\vec{v_i}$, $\vec{v_l}$) and the vector representation of t in test message ($\vec{v_u}$), now we measure the similarity between two tweets u_r and u_s. For each k-th index word w_k in u_r and l-th index word w_l in u_s we compute the cosine similarity between the embedded vectors of the words and fill up a similarity matrix M. We select the matrix cell that has the highest similarity, add this similarity score to the total similarity Sim, remove the row and column from M that has highest similarity score, and repeat the procedure (similar to Algorithm 1). We noticed that $MVME_{we}$ algorithm carefully chooses the best candidate word w_l in u_s

Expr.	Senses	Avg. P	Avg. R	Avg. F1	Max. F1(Target)	Min. F1(Target)
$PPMI_{bl}$	I	73.5 ± 3.6	84.6 ± 6.0	78.5 ± 3.2	83.9(mature)	68.8(wonder)
	L	83.1 ± 5.0	70.6 ± 5.5	76.1 ± 3.4	82.7(love)	68.3(nice)
	I	67.8 ± 7.0	76.2 ± 13.6	70.4 ± 7.6	81.8(joy)	43.8(like)
	L_{sent}	74.2 ± 7.1	62.7 ± 12.8	66.9 ± 6.6	78.6(joy)	47.1(interested)
WTMF	I	83.0 ± 3.4	87.2 ± 5.4	84.9 ± 2.4	91.4(mature)	78.7(wonder)
	L	87.5 ± 4.4	82.7 ± 4.5	84.9 ± 2.2	90.5(mature)	80.6(nice)
	I	67.4 ± 5.5	86.5 ± 5.1	75.6 ± 3.9	84.4(joy)	65.8(interested)
	L_{sent}	82.1 ± 5.8	58.9 ± 9.7	68.1 ± 7.2	81.5(joy)	50.0(genius)
GloVe	I	83.7 ± 3.6	85.6 ± 5.6	84.5 ± 2.8	90.6(joy)	78.8(sweet)
	L	86.3 ± 4.6	84.0 ± 4.3	85.0 ± 2.5	89.6(joy)	79.2(like)
	I	70.7 ± 5.1	84.3 ± 5.0	76.8 ± 3.9	85.4(joy)	67.1(interested)
	L_{sent}	80.7 ± 5.4	64.7 ± 8.5	71.5 ± 6.1	84.0(joy)	54.7(hot)
$w2v_{sg}$	I	84.9 ± 3.3	87.0 ± 4.8	85.8 ± 2.6	90.9(mature)	80.7(like)
	L	87.5 ± 4.1	85.1 ± 4.0	86.2 ± 2.5	90.7(mature)	80.2(like)
	I	70.8 ± 4.8	85.7 ± 5.1	77.4 ± 4.0	86.7(joy)	68.1(interested)
	L_{sent}	82.2 ± 5.7	64.3 ± 7.8	71.9 ± 5.9	85.4(joy)	57.4(interested)
$w2v_{cbow}$	I	84.9 ± 3.2	86.7 ± 4.7	85.6 ± 2.5	90.9(mature)	80.7(like)
	L	87.3 ± 4.0	85.1 ± 3.8	86.1 ± 2.4	90.7(mature)	80.2(like)
	I	70.7 ± 4.8	85.8 ± 5.0	77.4 ± 4.0	86.4(joy)	68.6(attractive)
	L_{sent}	82.0 ± 5.6	64.0 ± 7.7	71.7 ± 5.8	85.0(joy)	58.7(interested)

Table 3.12: Evaluation of distributional approaches (PMI and word embedding) for LISD experiments

for the w_k word in u_r since w_l is the most similar word to w_k. The algorithm continues the same procedure for all the remaining words in u_r and u_s. The final Sim is used as the kernel similarity between u_r and u_s. We augment this kernel $kernel_{we}$ into libSVM tool and during evaluation we run supervised LISD classification for each target word t separately.

Results and Discussion: Tables 3.12 and 3.13 show the results for the LISD experiments using the distributional approaches and classification-based approaches. For brevity, we only report the average Precision (P), Recall (R), and F1 scores with their standard deviation (SD) (given by "±"), and the targets with maximum/minimum F1 scores. $w2v_{sg}$ and $w2v_{cbow}$ represent the skip-gram and CBOW models implemented in word2vec, respectively.

Table 3.12 presents the results of distributional approaches. We observe that the word embedding methods have better performance than the PPMI baseline for both I vs. L and I vs. L_{sent} disambiguation tasks. Also, the average P/R/F1 scores for I vs. L are much higher than for I vs. L_{sent}. Since all tweets with L_{sent} sense were collected

Expr.	Senses	Avg. P	Avg. R	Avg. F1	Max.	Min.
					F1(Target)	F1(Target)
SVM_{bl}	I	87.0 ± 3.3	85.6 ± 3.1	86.3 ± 2.7	91.7(yeah)	75.4(sweet)
	L	85.9 ± 2.8	87.1 ± 3.6	86.5 ± 2.8	91.8(yeah)	76.1(sweet)
	I	77.3 ± 4.6	78.2 ± 4.2	77.7 ± 3.8	85.5(love)	68.6(brilliant)
	L_{sent}	77.8 ± 3.7	76.7 ± 6.4	77.1 ± 4.7	85.8(love)	64.6(attractive)
k_{WTMF}	I	94.1 ± 2.2	94.6 ± 1.8	94.3 ± 1.8	97.3(brilliant)	88.3(joy)
	L	94.6 ± 1.8	94.0 ± 2.3	94.3 ± 1.9	97.2(mature)	87.9(joy)
	I	79.0 ± 4.6	78.8 ± 4.4	78.8 ± 3.8	84.8(mature)	61.0(genius)
	L_{sent}	78.8 ± 3.7	78.9 ± 4.9	78.8 ± 3.6	85.4(mature)	63.5(genius)
k_{GloVe}	I	95.7 ± 1.6	97.4 ± 1.7	96.5 ± 1.1	99.1(mature)	92.9(glad)
	L	97.4 ± 1.6	95.6 ± 1.7	96.5 ± 1.2	99.1(mature)	92.7(interested)
	I	79.5 ± 3.5	83.1 ± 3.0	81.2 ± 2.8	86.9(joy)	74.2(attractive)
	L_{sent}	82.2 ± 3.0	78.3 ± 4.4	80.2 ± 3.4	86.6(joy)	69.2(attractive)
$k_{w2v_{sg}}$	I	96.6 ± 1.1	98.5 ± 0.6	97.5 ± 0.4	99.2(cute)	93.8(interested)
	L	98.5 ± 0.7	96.5 ± 1.2	97.5 ± 0.5	99.2(cute)	93.5(interested)
	I	81.9 ± 3.8	88.1 ± 3.2	84.8 ± 3.0	88.8(love)	74.2(genius)
	L_{sent}	87.0 ± 3.2	80.2 ± 4.7	83.4 ± 3.5	88.8(love)	73.3(genius)
$k_{w2v_{cbow}}$	I	96.4 ± 1.0	98.2 ± 1.1	97.3 ± 0.6	99.1(mature)	93.8(interested)
	L	98.2 ± 1.1	96.3 ± 1.1	97.2 ± 0.7	99.1(mature)	93.5(interested)
	I	81.7 ± 3.8	88.6 ± 2.9	84.9 ± 2.8	89.5(love)	74.8(genius)
	L_{sent}	87.4 ± 2.9	79.9 ± 4.8	83.4 ± 3.4	89.2(love)	74.4(genius)

Table 3.13: Evaluation of classification approaches (SVM_{bl} and $kernel_{we}$) for LISD experiments. $kernel_{we}$s are abbreviated as k_{we} for brevity.

using sentiment hashtags (González-Ibáñez et al., 2011), they might be lexically more similar to the I tweets than the L tweets are and thus identifying the sense of a target word t between I vs. L_{sent} is a harder task. In Table 3.12 we also observe that the average F1 scores between WTMF, $w2v_{sg}$, $w2v_{cbow}$, and GloVe are comparable and between 84%-86%, with $w2v_{sg}$ and $w2v_{cbow}$ achieving slightly higher F1.

Table 3.13 outlines the LISD experiments using the classification approaches: SVM baseline (SVM_{bl}) and SVM using the $kernel_{we}$ with word embeddings ($kernel_{WTMF}$, $kernel_{GloVe}$, $kernel_{w2v_{sg}}$, and $kernel_{w2v_{cbow}}$). The classification approaches give better performance compared to the distributional approaches. The SVM_{bl} is around 7-8 % higher than the $PPMI_{bl}$ and comparable with the word embeddings used in distributional approaches (Table 3.12).

In addition, our new SVM kernel method using word embeddings shows significantly better results when compared to the SVM_{bl} (and distributional approaches). For instance, for the I vs. L task, the average F1 is 96-97%, which is more than 10%

higher than SVM_{bl}. Similarly, for I 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 SVM_{bl}, 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 $kernel_{we}$ 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. Similar to Table 3.12, here, the average F1 scores for I vs. L task are higher than the I vs. L_{sent} results.

In terms of the best and worst performing targets, SVM_{bl} prefers targets with more training data (e.g., "yeah", "love" vs. "sweet", "attractive"; see Table 3.10). 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 3.12 and 3.13). This can be explained by the fact that for words, such as "joy", "mature", "cute", and "brilliant", the co-texts of their literal and ironic sense are quite different, and DSMs and word embeddings are able to capture the difference. For example, observe in the Table 3.11, negative sentiment words, i.e., "sick", "working", "snow" are the context words for targets "joy" and "love", where as positive sentiment words, such as, "blessed", "family", "christmas", and "peace" are the context words for I or L_{sent} senses. Overall, out of 37 targets, only 5 targets ("mature", "joy", "cute", "love", and "yeah") achieved "maximum" F1 scores in various experimental settings (Tables 3.12 and 3.13) 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%). For distributional approaches, Standard Deviation (SD) is slightly higher (5-8%) for several cases.

3.5.2.2 Identifying Incongruence at the Utterance Level

In the current section, we address whether we can identify the incongruence in the utterance itself. Later, in Section 3.6 we present our research in using conversation context for verbal irony identification and show how incongruence can be detected between a ironic utterance and its prior comments in a conversation. We address the following research question here,

- RQ5: Does modeling semantic incongruence assists in verbal irony detection at utterance level?

Riloff et al. (2013) have first shown that a common form of verbal irony on Twitter consists of a *positive* sentiment contrasted with a *negative situation*. For example, many ironic tweets include a positive sentiment, such as“love” or “enjoy”, followed by an expression that describes an undesirable activity or state (e.g., “going to the hospital’ or “being ignored”). They used a bootstrapping algorithm to identify the positive sentiment words and n-grams representing the negative situations and used these as features in verbal irony detection. However, the recall was low and later Joshi et al. (2015) improved the accuracy while combining both positive/negative and negative/positive contrast between a sentiment and a situation.

To detect the contrast (i.e., semantic incongruence) between the sentiment and the situation in an ironic utterance we continue working on the same framework of reversal of valence identification that was introduced in the previous section. This setup enables a clear identification of the “target” words (words that can have a literal or a ironic sense) and allows us to model the relation between the target words and their co-text in order to predict whether the target word is used in a ironic or literal sense. Thus, instead of applying bootstrapping algorithm to detect positive/negative sentiment and situation (similar to Riloff et al. (2013)), we directly use all the target words (from the LISD experiments in the previous section) and their situations to predict the incongruence.

We employ convolutional neural networks (CNN) (LeCun and Bengio, 1995) to model incongruence between the sentiment and the situation. CNN is a type of feed-forward artificial neural network that exploits spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. Although, originally developed for image recognitions, CNNs have recently been shown to achieve impressive results on various tasks that can be modeled as sentence classification (e.g., sentiment analysis, question type classification) (Kim, 2014; Kalchbrenner et al., 2014; Lei et al., 2015). In NLP problems, the input to CNNs are usually n-grams (e.g., a trigram or four-gram) but it could be characters too. Typically, the n-grams are presented by their word embeddings (low-dimensional representations) like word2vec or GloVe (Mikolov et al., 2013a; Pennington et al., 2014). CNN filters slide over the n-grams where the width of the filters are usually same as the width of the n-grams. Note, although CNNs do not care about the position of a word in a sentence unlike the Recurrent Neural Networks (Mikolov et al., 2010), they perform reasonably well for classification, especially for short texts such as tweets.[13] Another argument in favor of using CNNs is training process if much faster than recurrent networks. CNN directly enables us to discover whether there is any incongruity between a target and their co-text. Our model is based on the use of a dual-CNN architecture. The first CNN is applied over the target word and its direct neighbor. The second CNN is applied over the rest of the utterance (i.e., the ironic situation). We reckon such explicit and generic modeling of the n-gram information between sentiment and ironic situation will help in identifying the incongruence. We utilize the same training/test corpus for all the target words (i.e., 37 target words) introduced in the LSSD section. Below, we formally introduce the model.

We propose a modification of Kim (2014) one-layer CNN (shown in Figure 3.3), subsequently denoted as CNN_T. Given a tweet u of n words let $x_i \in \mathbb{R}^d$ represents the d dimensional word representation of the i^{th} word u. A convolution operation involves filters $\mathbf{w} \in \mathbb{R}^{\mathbf{kd}}$, which is applied to a window of k words to produce a new feature. For instance, a feature f_i can be generated from the words $x_{i:i+k-1}$ by

$$f_i = c(\mathbf{w}.x_{i:i+k-1} + b) \tag{3.1}$$

Here $b \in \mathbb{R}$ is a bias term and c is a non-linear function such as a Rectified Linear Unit (ReLU). This filter is applied to each window of words to generate a feature map.

The concatenation of k words ($\oplus$ is the concatenation operator) is represented as

[13] This is comparable to the idea of using Bag of Words model for text classification. The n-grams are treated independently without emphasizing their positions, but has nonetheless been the standard baseline approach for years in NLP research.

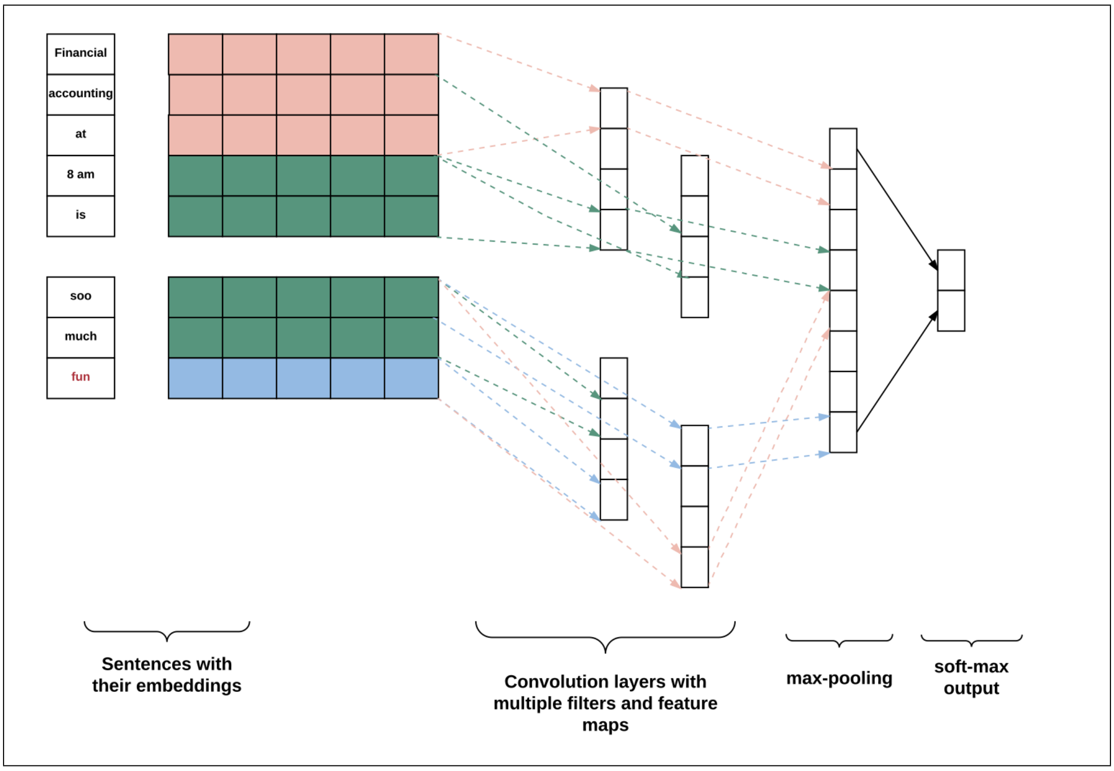

Financial
accounting
at
8 am
is
soo
much
fun
Sentences with their embeddings
Convolution layers with multiple filters and feature maps
max-pooling
soft-max output

Figure 3.3: Example of the CNN_T architecture (Figure is inspired by Kim (2014))

follows:

$$x_{1:k} = [x_1 \oplus x_2 \oplus \cdots \oplus x_k] \tag{3.2}$$

Lets assume u contains a target word x_t, then we propose an architecture with two CNNs with the goal of explicitly modeling the relation between the target word x_t and its co-text. In this architecture, shown in the Figure 3.3 we use two separate CNNs; one performs convolution operations over x_t and its immediate neighboring words and the other CNN operates convolutions over the remaining words in u (i.e., the co-text of x_t).[14] We use very simple heuristics to select the text for the first CNN; (a) if the target word x_t appears in the first three words or the last three words of the utterance, then respectively $x_1 \oplus x_2 \oplus x_3$ or $x_{n-2} \oplus x_{n-1} \oplus x_n$ represent the CNN that performs convolution operations over x_t. (b) Otherwise, we consider the trigrams that contain the target x_t (i.e., $[x_{t-1} \oplus x_t \oplus x_{t+1}]$). Remaining words in the utterance are operated by the second CNN.

[14] The use of two or more identical subnetworks is sometime denoted as the “siamese network” (Koch et al., 2015).

Consider the Figure 3.3 that represents the CNN_T architecture on the ironic utterance - “financial accounting at 8 am is soo much fun”. The lower CNN performs convolution operations over “soo much fun” (i.e., the last three words of the utterance represented by $x_{n-2} \oplus x_{n-1} \oplus x_n$, we do not model the hashtag label). The second CNN has as input the rest of utterance (i.e., “financial accounting at 8 am is”). We separately apply max-pooling operation to each feature maps generated from the convolution operations in the two CNNs and then concatenate the maximum valued features from the two CNNs for the softmax layer. We use three separate filters of unigram, bigram, and trigrams and applied zero-padding when necessary.

Parameters and pre-trained word vectors: We select rectified linear units (ReLU) as the non-linear operator, filter windows with 100 feature maps for each unigram, bigram, and trigram filters, dropout rate of 0.5, and mini-batch size of 25 (similar to Kim (2014)). We use the word-embedding model (Skip-gram model in *word2vec* tool (Mikolov et al., 2013a)) that was used in the LSSD experiment (Section 3.5.2.1). We kept the same settings used (100-d vectors; 10 word context window).

Results and Discussions:

We present the results in the Table 3.14. SVM_{bl} is a strong BoW baseline based on binary morpho-syntactic and typography features (i.e., n-grams, sentiment lexicons, interjections, tag questions, emoticons, the same baseline reported in the LSSD experiment). $k_{w2v_{cbow}}$ is the best SVM kernel (with cbow-embedding) that we designed in the LSSD experiment (Section 3.5.2.1). CNN_{Kim} is the one-layer CNN (filter windows of 1, 2, 3 with 100 feature maps) of Kim (2014), and CNN_{Lei} is the non-linear, non-consecutive CNN with default parameters (Lei et al., 2015).

For brevity, we only report the average Precision (P), Recall (R), and F1 scores with their standard deviation (given by ‘±’), and the targets with maximum/minimum F1 scores. We notice for both I and L_{sent} category, CNN_T performs best with around 2% F1 improvement over other CNN (CNN_{Kim}) and around 4% F1 improvement over word embeddings kernel ($k_{w2v_{cbow}}$). The target-centric CNN shows highest performance for target words that have considerable less training data (<2K instances per class). While the performance of all CNNs models are similar, error analysis highlighted an interesting qualitative difference between the CNN_T and the baseline CNN models (we compared against different CNN models proposed by (Kim, 2014) and (Lei et al., 2015)). We observe that when the verbal irony is characterized by co-text incongruity the target-centric CNNs are more accurate in predicting the correct class, especially where the incongruity is indirect (i.e., "*it **cool** ... nothing better than wait in parking lot for brother*", "*spending your saturday night in bed is always the **best***"; target words are **bold**). Comparing the target-centric CNNs to SVM_{we} we found that the former is more precise when the target appears at the end ("*financial accounting is sooooo much **fun** at 8am*", "*leaving homecoming early to be ill ... was **fantastic***"). CNN models are superior in capturing the sequential information whereas SVM_{we} algorithm runs an alignment between words without capturing the order of words. We also experimented with different settings, such as we keep the embedding vectors as non-static (i.e., letting the word-vectors to update during the training). However, we do not observe any significant difference in performance.

In this research using the CNN_T we observe often the model fails where the "incongruity" is beyond the boundary of the utterance. However, many times in these case, users tend to use *irony markers* to express the verbal irony and we notice even though the CNN models are unable to identify verbal irony, discrete feature based methods can identify the verbal irony. Particularly in Twitter, where the scope of expressing the context is limited (140 chars), users try to overcome the limitation by using emoticons, multiple interjections, alternate spellings (observe the use of multiple "!" and the emoticon ":(" for *co-text incongruity*. In future we plan to combine models that capture irony markers (i.e., discrete features) and irony factors (i.e., deep learning methods) to advance the state-of-the-art in verbal irony detection.

Until this section we have described utterance-based models. In the next section, we present our research where we model both utterance and the local conversation context to detect verbal irony.

Expr.	Senses	Avg. P	Avg. R	Avg. F1	Max. F1(Target)	Min. F1(Target)
SVM_{bl}	I	77.3 ± 4.6	78.2 ± 4.2	77.7 ± 3.8	85.5(love)	68.6(brilliant)
	L_{sent}	77.8 ± 3.7	76.7 ± 6.4	77.1 ± 4.7	85.8(love)	64.6(attractive)
$k_{w2v_{cbow}}$	I	81.7 ± 3.8	88.6 ± 2.9	84.9 ± 2.8	89.5(love)	74.8(genius)
	L_{sent}	87.4 ± 2.9	79.9 ± 4.8	83.4 ± 3.4	89.2(love)	74.4(genius)
CNN_{Kim}	I	88.3 ± 7.2	86.6 ± 17.3	87.3 ± 6.1	93.3(shocked)	72.9(genius)
	L_{sent}	85.9 ± 7.2	87.3 ± 17.3	86.5 ± 6.1	92.9(love)	71.1(genius)
CNN_{Lei}	I	78.8 ± 8.9	85.2 ± 5.6	81.5 ± 5.3	92.2(love)	70.8(attractive)
	L_{sent}	83.6 ± 4.9	75.4 ± 14.3	78.6 ± 9.7	92.1(love)	51.7(interested)
CNN_T	I	**89.8** ± 7.1	88.4 ± 7.9	**89.0** ± 5.3	94.4(love)	75.4(genius)
	L_{sent}	87.5 ± 7.4	**88.9** ± 8.3	**88.4** ± 5.0	94.3(love)	69.7(genius)

Table 3.14: Evaluation of classification approaches for sense-disambiguation experiments (highest scores are **bold**)

Platform	Turn Type	Turn pairs
Twiter	P_TURN	**userA:** Plane window shades are open during take-off & landing so that people can see if there is fire in case of an accident URL .
	C_TURN	**userB**: @UserA awesome !! one more reason to feel really great about flying . . . #sarcasm.

Table 3.15: Ironic turns (C_TURN) and their respective prior turns (P_TURN) in Twitter and discussion forums.

3.6 Role of Conversation Context in Verbal Irony Identification

Our analysis of irony markers and factors has been based on the modeling of utterances in isolation. However, it has been argued that verbal irony, is a type of interactional phenomenon with specific perlocutionary effects on the hearer (Haverkate, 1990), such as to break their pattern of expectation. Thus, to be able to detect speakers' ironic intent it is necessary (even if maybe not sufficient) to consider their utterances in the larger conversation context. Consider the Twitter conversation example in Table 3.15. Without the context of UserA's statement, the ironic intent of UserB's response might not be detected.

In this section, we investigate the role of *conversation context* for the detection of verbal irony in social media discussions (Twitter conversations and discussion forums). The unit of analysis (i.e., what we label as ironic or non-ironic) is a message/turn in a social media conversation (i.e., a tweet in Twitter or a post/comment in discussion forums). We call this unit *current turn* (C_TURN). The conversation context that we consider is the *prior turn* (P_TURN), and when available also the *succeeding turn* (S_TURN), which is the reply to the *current turn*. Table 3.16 and Table 3.17 show examples from the IAC_{v2} corpus and $Reddit$ of ironic messages (C_TURNS; userB's post) and the conversation context given by the prior turn (P_TURN; userA' post) as well as the subsequent turn (S_TURN; userC's post, only in Table 3.16), respectively.

We address three specific research questions in this section:

- RQ6: Does modeling of conversation context help in verbal irony detection?
- RQ7: Can humans and computational models identify what part of the prior turn (P_TURN) triggered the ironic reply (C_TURN) (e.g., which sentence(s) from userC's turn triggered userD's ironic reply in Table 2.5.)?
- RQ8: Given a ironic message (C_TURN) that contains multiple sentences, can humans and computational models identify the specific sentence that is ironic?

To answer the sixth RQ, we investigate both Support Vector Machine models (Chang and Lin, 2011) with linguistically-motivated discrete features and several types of Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) that can model both conversation context (i.e., P_TURN, S_TURN or both) and current turn (C_TURN). LSTM networks (formally introduced in the next section) are sequential networks that are able to learn long-term dependencies (Hochreiter and Schmidhuber, 1997). The reasons for using LSTMs are threefold. First, it has been shown in related literature that LSTM networks are superior in semantic representation of sentences, especially capturing the long-distance dependency between words. Posts from discussion forums often contain multiple lines and LSTMs can maintain a hierarchical structure where words, sentences, and the full post can be represented by a single semantic vector. Second, LSTMs are also known as *gated* networks, meaning, in a sequence of *n* words, LSTM can control how much information to pass/block between the words. This is important because in discussion forums some posts are long and the model needs to identify the critical portions of the posts that are useful in irony detection. In the proposed architecture, each turn (i.e., context turn or the current turn) is represented by separate LSTM node. The output vectors from the nodes are concatenated and the model classifies that particular vector. In other words, the model learns how does the conversation context trigger an ironic turn. We also show that the conditional LSTM network (Rocktäschel et al., 2015) and LSTM networks with sentence level attention on current turn (C_TURN) *and* context (particularly the prior turn) outperform the LSTM model that reads only the current turn (C_TURN). Third and finally, LSTMs have been shown to be effective in Natural Language Inference (NLI) tasks such as Recognizing Textual Entailment, where the goal is to establish the *relationship* between two inputs (e.g., a premise and a hypothesis) (Bowman et al., 2015; Rocktäschel et al., 2015; Parikh et al., 2016)). Likewise, our aim is also determining the relationship between *context* and *response*, thus LSTM fits the goal of utilizing the contextual information. Finally, vanilla LSTMs can be modified into the "attention-based networks", which is useful for the qualitative research we conducted in Section 3.6.3.

Our computational models are tested on two different types of platforms: Twitter and discussion forums - IAC_{v2} and $Reddit$. We introduced all three corpora in Chapter 2 and Section 3.4 in detail. For IAC_{v2} corpus, of around 5 K ironic posts and their context are available whereas for $Reddit$ we are using 50 K posts. Both corpus are balanced between irony and non-irony categories. Note, since the discussion forum posts are not dependent on hashtags (i.e., sentiment hashtags) we do not make use of notations such as NI_{rand} or NI_{sent} as applicable to Twitter. Rather we denote any non-ironic utterances as NI and ironic utterances as I. The IAC_{v2} corpus contains only the prior turn as conversation context. Given that we are interested in studying also the subsequent turn as context, we checked to see whether for a current turn we can extract its subsequent turn from the general IAC corpus. Out of the 4,692 current turns, we found a total of 2,309 that have a subsequent turn (e.g., number of total subsequent turns is 2,786, since a candidate turn can have more than one subsequent reply) in the IAC corpus. We denote this corpus as IAC^{+}_{v2}. Examples from the IAC^{+}_{v2} are given in Table 3.16.

For $Twitter$ we observe that around 26 K of tweets are a reply to another tweet and

Turn Type	Social media discussion
P_TURN	**userA:** my State is going to heII in a handbasket since these lefties took over. emoticonXBanghead.
C_TURN	**userB:** Well since Bush took office the mantra has bee ”Local Control” has it not. Apparently the people of your state want whats happening. Local control in action. Rejoice in your victory.
S_TURN	**userC:** I think the trip was a constructive idea, especially for high risk middle school youths .... Perhaps the program didn’t respect their high risk homes enough. If it were a different group of students, the parents would have been told. The program was the YMCA, not lefty, but Christian based.
P_TURN	**userA:** In his early life, X had a reputation for drinking too much. Whether or not this affected his thinking is a question which should to be considered when asking questions about mormon theology ... emoticonXBanghead.
C_TURN	**userB:** Wow, that must be some good stuff he was drinking to keep him ’under the influence’ for THAT long!! :p
S_TURN	**userC:** Perhaps he was stoned on other drugs like the early writers of the bible.

Table 3.16: Ironic messages (C_TURNS) and their respective prior turns (P_TURN) and subsequent turns (S_TURN) from IAC_{v2}.

Turn Type	Social media discussion
P_TURN	**userA:** They’re (media) lying. The Uk, covers up islamic violence. Just pretend it’s not happening is the rule. Any who points out the truth will be arrested.
C_TURN	**userB:** But did they mention the gender of the attacker? No, because the media is covering the fact that it is 99of the times men that commit these crimes. #deportallmen

Table 3.17: Ironic messages (C_TURNS) and their respective prior turns (P_TURN) and subsequent turns (S_TURN) from $Reddit$.

thus our final Twitter conversations set contains 25,991 instances (12,215 instances for I class and 13,776 instances for the NI class).[15] We notice that 30% of the tweets have more than one tweet in the conversation context.

To address the seventh and the eighth RQs, 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 tasks (Section 3.6.3).

3.6.1 Computational Models and Experimental Setup

To answer the sixth research question "does modeling of conversation context help in verbal irony detection" we consider two binary classification tasks. As stated earlier, we refer to ironic instances as I and non-ironic instances as NI.

The first task is to predict whether the current turn (C_TURN abbreviated as ct) is ironic or not, considering it in isolation — I^{ct} vs. NI^{ct} task.

The second task is to predict whether the current turn is ironic or not, by taking into account both the current turn and its conversation context given by the prior turn (P_TURN, abbreviated as pt), succeeding turn (S_TURN, abbreviated as st) or both — $I^{ct+context}$ vs. $NI^{ct+context}$ task, where $context$ is pt, st or pt+st.

For all the corpora introduced in Section 3.4 — IAC_{v2}, IAC^{+}_{v2}, $Reddit$, and $Twitter$ — we conduct I^{ct} vs. NI^{ct} and I^{ct+pt} vs. NI^{ct+pt} classification tasks. For IAC^{+}_{v2} we also perform experiments considering the succeeding turn st as conversation context (i.e., I^{ct+st} vs. NI^{ct+st} and $I^{ct+pt+st}$ vs. $NI^{ct+pt+st}$).

SVM with discrete features (SVM$_{bl}$)

For baseline features, we used the same set of discrete features used in the LSSD experiment (Section 3.5.2.1) and identifying incongruity (Section 3.5.2.2). We used BoW features (i.e., unigram, bigram, and trigram representation of words), LIWC lexicons to represent pragmatic features, and irony markers as features (for irony marker features, see Section 3.5.1. For modeling conversation context, we introduce a new set of sentiment features that measure the similarities and dissimilarities in the context

[15] For tweets, this time we do not consider the NS_{rand} tweets.

and response. Here, two sentiment lexicons are also used to model the utterance sentiment: "MPQA" (Wilson et al., 2005) and "Opinion Lexicon" (Hu and Liu, 2004a). To capture sentiment, we count the number of positive and negative sentiment tokens, negations, and use a boolean feature that represents whether a reply contains both positive and negative sentiment tokens. For the I^{ct+pt} vs. NI^{ct+pt} classification task, we check whether the current turn ct has a different sentiment than the prior turn pt (similar to Joshi et al. (2015)). Given that ironic utterances often contain a positive sentiment towards a negative situation, we hypothesize that this feature will capture this type of sentiment incongruity.

Tokenization is conducted via CMU's Tweeboparser (Gimpel et al., 2011). For the discussion forum dataset we use the NLTK tool (Bird, 2006) for sentence boundary detection and tokenization. We used libSVM toolkit with Linear Kernel (Chang and Lin, 2011) with weights inversely proportional to the number of instances in each class.

Long Short-Term Memory Networks

LSTMs are a type of recurrent neural networks (RNNs) that are able to learn long-term dependencies (Hochreiter and Schmidhuber, 1997). LSTMs address the vanishing gradient problem commonly found in RNNs by incorporating gating functions into their state dynamics (Hochreiter and Schmidhuber, 1997). We introduce some notations and terminology standard in the LSTM literature (Tai et al., 2015). The LSTM unit at each time step t is defined as a collection of vectors: an input gate i_t, a forget gate f_t, an output gate o_t, a memory cell c_t, and a hidden state h_t. The LSTM transition equations are listed below:

$$\begin{aligned}
i_t &= \sigma(\mathbf{W}_i * [h_{t-1}, x_t] + b_i)\\
f_t &= \sigma(\mathbf{W}_f * [h_{t-1}, x_t] + b_f)\\
o_t &= \sigma(\mathbf{W}_o * [h_{t-1}, x_t] + b_o)\\
\tilde{C}_t &= \tanh(\mathbf{W}_c * [h_{t-1}, x_t] + b_c)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{C}_t\\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{3.3}$$

where x_t is the input at the current time step, σ is the logistic sigmoid function and $\odot$ denotes element-wise multiplication. The input gate controls how much each unit is updated, the forget gate controls the extent to which the previous memory cell is forgotten, and the output gate controls the exposure of the internal memory state. The hidden state vector is a gated, partial view of the state of the unit's internal memory cell. Since the value of the gating variables vary for each vector element, the model can learn to represent information over multiple time scales.

Since our goal is to explore the role of contextual information (e.g., prior turn and/or succeeding turn) for recognizing whether the current turn is ironic or not, we will use multiple LSTMs: one which reads the current turn and one (or two) which read(s) the context (e.g., one LSTM will read the prior turn and one will read the succeeding turn when available).

Attention-based LSTM Networks

Attentive neural networks have been shown to perform well on a variety of NLP tasks (Yang et al., 2016; Yin et al., 2015; Xu et al., 2015). Using attention-based LSTM will accomplish two goals: (1) test whether they achieve higher performance than simple LSTM models and (2) use the attention weights produced by the LSTM models to perform the qualitative analyses that enable us to answer the last two questions we want to address (e.g., which portions of context triggers the ironic reply, used to answer RQ7 and RQ8).

Yang et al. (2016) have included two levels of attention mechanisms, one at the word level and another at the sentence level where the sentences are in turn produced by attentions over words (i.e., the hierarchical model). We experiment with two architectures: one hierarchical that uses both word-level and sentence level attention (Yang et al., 2016), and one which uses only sentence-level attention (here we use only the average word embeddings to represent the sentences).

One question we want to address is whether sentence level attention weights indicate what sentence(s) in prior turn trigger(s) the ironic reply. In the discussion forum datasets, prior turns are usually more than three sentences long and thus attention weights could indicate what part of the prior turn triggers the ironic post ct.

Figure 3.4 shows the high-level structure of the model where the conversation context is represented by the prior turn pt. The context (left) is read by an LSTM ($LSTM_{pt}$) whereas the current turn ct (right) is read by another LSTM ($LSTM_{ct}$). Note, for the model where we consider the succeeding turn st as well, we simply use another LSTM to read st. For brevity we only show the sentence-level attention.

Let the context pt contain d sentences and each sentence s_{pt_i} contain T_{pt_i} words. Similar to the notation of (Yang et al., 2016), we first feed the sentence annotation h_{pt_i} through a one layer MLP to get u_{pt_i} as a hidden representation of h_{pt_i}, then we weight the sentence u_{pt_i} by measuring similarity with a sentence level context vector u_{pt_s}. This gives a normalized importance weight α_{pt_i} through a softmax function. v_{pt} is the vector that summarize all the information of sentences in the context ($LSTM_{pt}$).

$$v_{pt} = \sum_{i \in [1,d]} \alpha_{pt_i} h_{pt_i} \tag{3.4}$$

where attention is calculated as:

$$\alpha_{pt_i} = \frac{\exp(u_{pt_i}^T u_{pt_s})}{\sum_{i \in [1,d]} \exp(u_{pt_i}^T u_{pt_s})} \tag{3.5}$$

Likewise we compute v_{ct} for the current turn ct via $LSTM_{ct}$ (similar to equation 3.4; also shown in Figure 3.4). Finally, we concatenate the vector v_{pt} and v_{ct} from the two LSTMs for the final softmax decision (i.e., predicting the I or NI class). In case of using the succeeding turn st also in the model, we concatenate the vectors v_{pt}, v_{ct} and v_{st}.

As stated earlier in this section, we also experiment with both word and sentence level attentions in a hierarchical fashion similarly to the approach proposed by Yang et al. (2016). As we show in Section 3.6.2 however, we achieve best performance using just the sentence-level attention. A possible explanation is that attention over both words and sentences seek to learn a large number of model parameters and given the moderate size of the discussion forum corpora they might overfit.

For tweets, we treat each individual tweet as a sentence. The majority of tweets consist of a single sentence and even if there are multiple sentences in a tweet, often

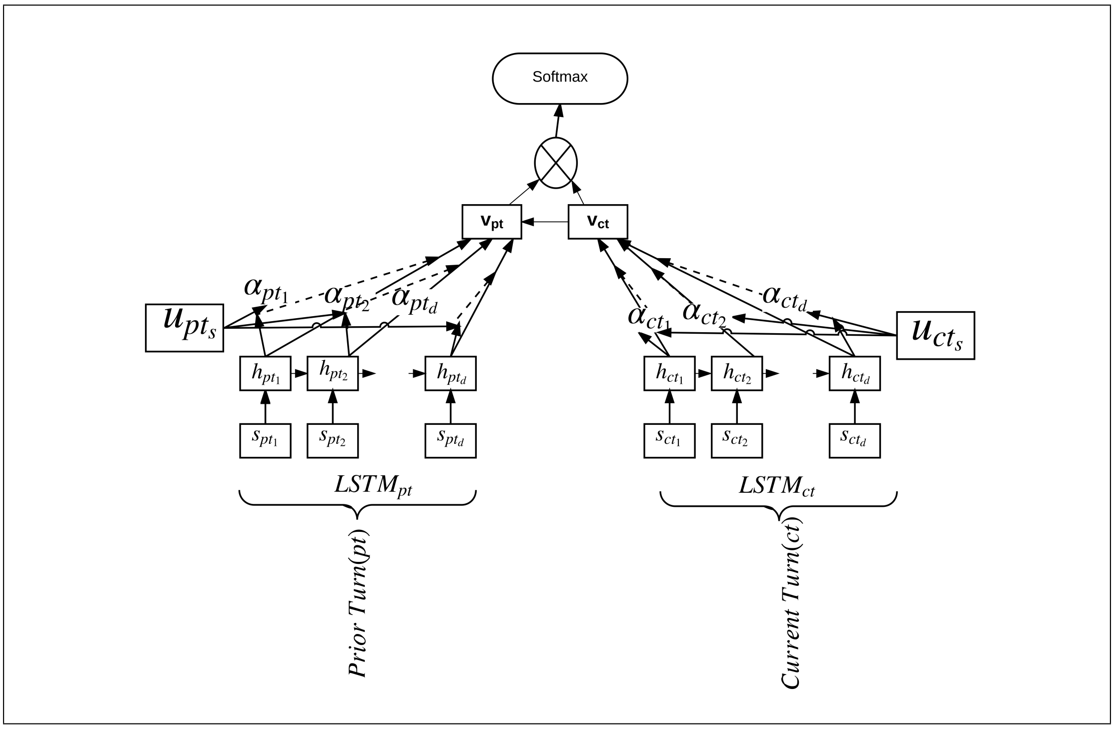


Figure 3.4: Sentence-level Attention Network for prior turn pt and current turn ct. Figure is inspired by Yang et al. (2016)

one sentence contains only hashtags, URLs, and emoticons making them uninformative if treated in isolation.

Conditional LSTM Networks 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 – $LSTM_{pt}$ and $LSTM_{ct}$ – similar to the previous architecture without any attention, but for $LSTM_{ct}$, its memory state is initialized with the last cell state of $LSTM_{pt}$. In other words, $LSTM_{ct}$ is conditioned on the representation of the $LSTM_{pt}$ that is built on the prior turn pt. For models that use the successive turn st as the context the LSTM representation $LSTM_{st}$ is conditioned on the representation of the $LSTM_{ct}$. Figure 3.5 shows the model where we consider the current turn ct is conditioned on the prior turn pt.

Parameters and pre-trained word vectors For both discussion forum and $Twitter$, we split randomly the corpus into training (80%), development (10%), and test (10%), maintaining the same distribution of ironic vs. non-ironic data in training, development and test. For Twitter we used the skip-gram word-embeddings (100-dimension) used in

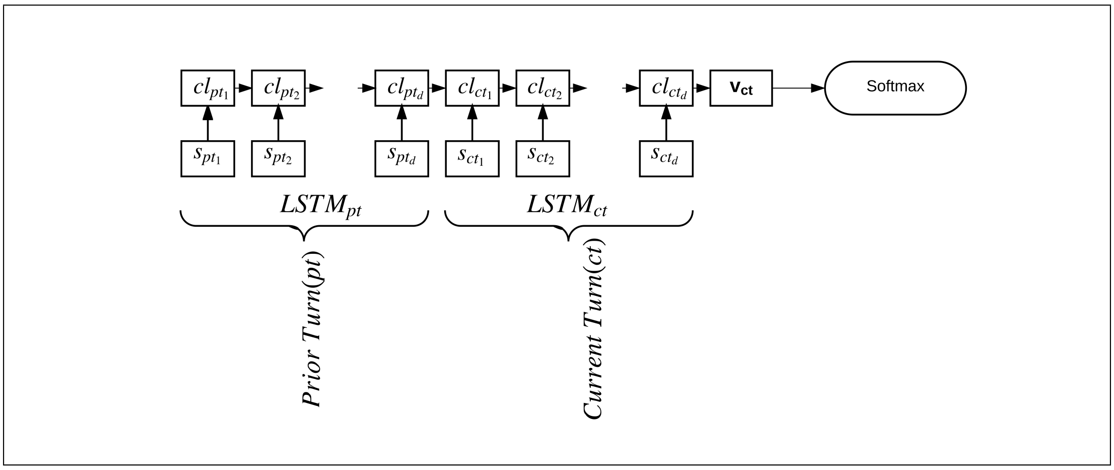


Figure 3.5: Conditional LSTM Network for prior turn pt and current turn ct; Figure is inspired by the model proposed in Rocktäschel et al. (2015)

the LSSD experiments that was built using over 2.5 million tweets.[16] For discussion forums, we use the standard Google n-gram $word2vec$ pre-trained model (300-dimension) (Mikolov et al., 2013a). We do not optimize the word embedding during training. Out-of-vocabulary words in the training set are randomly initialized via sampling values uniformly from (-0.05,0.05). We use the development data to tune the parameters and selected dropout rate of 0.5 (from [.25,0.5, 0.75]), L_2 regularization strength and evaluate only that configuration on the test set. For both datasets mini-batch size of 16 is employed.

3.6.2 Results and Discussion

In this section we present a quantitative analysis aimed at addressing our first RQ‘ “does modeling conversation context help in verbal irony detection?” First, we consider just the *prior turn as conversation context* and show results of our various models on all datasets: IAC_{v2}, $Reddit$ and $Twitter$ (Section 3.6.2.1). In addition, we perform an experiment where we train on $Reddit$ (discussion forum, self-labeled) and test on IAC_{v2} (discussion forum, labeled via crowdsourcing). Second, we consider *both prior turn and the succeeding turn as context*, and report results of various models on our IAC^{+}_{v2} dataset (Section 3.6.2.2). We report Precision (P), Recall (R), and F1 scores on

[16] https://github.com/debanjanghosh/sarcasm_wsd

Experiment	I			NI		
	P	R	F1	P	R	F1
SVM$_{bl}^{ct}$	65.55	66.67	66.10	66.10	64.96	65.52
SVM$_{bl}^{ct+pt}$	63.32	61.97	62.63	62.77	64.10	63.5
LSTM^{ct}	67.90	66.23	67.1	67.08	**68.80**	67.93
LSTM^{ct}+LSTM^{pt}	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$^{ct_{a_s}}$	69.45	70.94	70.19	70.30	68.80	69.45
LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$	66.90	**82.05**	**73.70**	**76.80**	59.40	66.99
LSTM$^{ct_{a_{w+s}}}$+LSTM$^{pt_{a_{w+s}}}$	65.90	74.35	69.88	70.59	61.53	65.75

Table 3.18: Experimental results for the discussion forum dataset (IAC_{v2}) (**bold** are best scores)

ironic (I) and non-ironic (NI) classes.

3.6.2.1 Prior Turn as Conversation Context

SVM$_{bl}^{ct}$ and SVM$_{bl}^{ct+pt}$ represent the performance of the SVM models with discrete features when using only the current turn ct and the ct together with the prior turn pt, respectively. LSTM^{ct} and LSTM^{ct+pt} represent the performance of the simple LSTM models when using only the current turn ct and the ct together with the prior turn pt, respectively. LSTM^{pt_a} and LSTM^{ct_a} are the attention-based LSTM models of context pt and current turn ct, 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 that conditions the LSTM that reads the current turn on the LSTM that reads the prior turn (no attention). Given these notations we present the results on each of the three datasets.

IAC_{v2} ***corpus:*** Table 3.18 shows the classification results on the IAC_{v2} dataset. Although a vast majority of the prior turn posts contain 3-4 sentences, around 100 have more than ten sentences and thus we set a cutoff to a maximum of ten sentences for context modeling. For the current turn ct we consider the entire post.

The SVM_{bl} models that are based on discrete features did not perform very well, and adding the context of the prior turn pt actually hurt the performance. Regarding the performance of the neural network models, we observe that modeling prior turn pt as context improves the performance using all types of LSTM architectures that read both context (pt) and current turn(ct) (results are statistically significant when compared to LSTM^{ct}). The highest performance when considering both the I and NI classes is achieved by the LSTM^{conditional} model (73.32% F1 for I class and 70.56% F1 for NI, showing a 6% and 3% improvement over LSTM^{ct} for I and NI classes, respectively). The LSTM model with sentence-level attentions on both context and current turn (LSTM$^{pt_{a_s}}$+LSTM$^{ct_{a_s}}$) gives the best F1 score of 73.7% for the I class. For the NI class, while we notice an improvement in precision we notice a drop in recall when compared to the LSTM model with sentence level attention only on the current post (LSTM$^{ct_{a_s}}$). Remember that sentence-level attentions are based on average word embeddings. We also experimented with the hierarchical attention model where each sentence is represented by a *weighted average* of its word embeddings. In this case, attentions are based on words and sentences and we follow the architecture of hierarchical attention network (Yang et al., 2016). We observe that the performance (69.88% F1 for I category) deteriorates, probably due to the lack of enough training data. Since attention over words and sentences (together) seek to learn more model parameters, adding more training data will be helpful. For the $Reddit$ and $Twitter$ data (see below), these models may become better, but still not on par with just sentence level attention showing that even larger datasets might be needed.

Twitter Corpus Table 3.19 shows the results on the Twitter dataset. As for IAC_{v2}, adding context using the SVM models does not show a statistically significant improvement. For the neural networks models, similar to the results on the IAC_{v2} dataset, the LSTM models that read both context and current turn outperform the LSTM model that reads only the current turn (LSTM^{ct}). The best performing architectures are again the LSTM^{conditional} and LSTM with sentence-level attentions (LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$). LSTM^{conditional} model shows an improvement of 11% F1 on the I class and 4-5%F1 on the NI class, compared to LSTM^{ct}. For the attention-based models, the improvement using context is smaller ($\sim$2% F1). We kept the maximum length of prior tweets to the last five tweets in the conversation context, when available. We also considered experiment with only the "last" tweet (i.e., LSTM$^{ct_{a_s}}$+LSTM$^{last_pt_{a_s}}$). We observe although the F1 for the non-ironic category is high (76%), for the ironic category it is

Experiment	I			NI		
	P	R	F1	P	R	F1
SVM$_{bl}^{ct}$	64.20	64.95	64.57	69.0	68.30	68.7
SVM$_{bl}^{ct+pt}$	65.64	65.86	65.75	70.11	69.91	70.0
LSTM^{ct}	73.25	58.72	65.19	61.47	75.44	67.74
LSTM^{ct}+LSTM^{pt}	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$^{ct_{a_s}}$	76.00	73.18	74.56	70.52	73.52	71.9
LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$	**77.25**	75.51	**76.36**	72.65	**74.52**	**73.57**
LSTM$^{ct_{a_s}}$+LSTM$^{last\_pt_{a_s}}$	73.10	69.69	71.36	74.58	**77.62**	**76.07**
LSTM$^{ct_{a_w}}$+LSTM$^{pt_{a_w}}$	76.74	69.77	73.09	68.63	75.77	72.02
LSTM$^{ct_{a_{w+s}}}$+LSTM$^{pt_{a_{w+s}}}$	76.42	71.37	73.81	69.50	74.77	72.04

Table 3.19: Experimental results for Twitter dataset (**bold** are best scores)

low (e.g., 71.3%). This shows that considering a larger conversation context of multiple prior turns rather than just the last prior turn could assists in achieving higher accuracy particularly in Twitter where each turn/tweet is short.

Reddit Corpus Table 3.20 shows the results of the experiments on Reddit data. There are two major differences between this corpus and the IAC_{v2} corpus. First, since the original release of the Reddit corpus (Khodak et al., 2017) is very large, we select a subcorpus that is much largers than the IAC_{v2} data containing 50K instances. In addition, we selected posts (both pt and ct) that consist of a maximum of seven sentences primarily to be comparable with the IAC_{v2} data.[17] Second, unlike the IAC_{v2} corpus, the ironic current turns ct are self-labeled so it is unknown whether there are any similarities between the nature of the data in the two discussion forums.

We observe that the SVM$_{bl}$ models perform similarly to the other discussion forum corpus IAC_{v2}. The SVM$_{bl}^{ct+pt}$ model performs poorly compared to the SVM$_{bl}^{ct}$ model. Similarly with the other datasets, adding the context of the prior turn when using the LSTM models helps. LSTM with sentence-level attention performs best for the ironic category. Except for the SVM$_{bl}$ we observe adding the prior turn pt helps in achieving higher performance. Note, Khodak et al. (2017) have evaluated the ironic utterances via BoW features and sentence embeddings and achieved an accuracy in mid 70%. However, they selected sentences between two to fifty words for the classification

[17] IAC_{v2} contains prior and current turns which contains mostly seven or fewer sentences.

Experiment		I			NI	
	P	R	F1	P	R	F1
SVM$_{bl}^{ct}$	72.54	72.92	72.73	72.77	72.4	72.56
SVM$_{bl}^{ct+pt}$	66.3	67.52	66.90	66.91	65.68	66.29
LSTM^{ct}	**81.29**	59.6	68.77	68.1	**86.28**	**76.12**
LSTM^{ct}+LSTM^{pt}	74.46	73.72	74.09	73.98	74.72	74.35
LSTM^{conditional}	73.72	71.6	72.64	72.40	74.48	73.42
LSTM$^{ct_{a_s}}$	74.87	74.28	74.58	74.48	75.08	**74.78**
LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$	73.11	**80.60**	**76.67**	**78.39**	70.36	74.16
LSTM$^{ct_{a_{w+s}}}$+LSTM$^{pt_{a_{w+s}}}$	74.50	74.68	**74.59**	74.62	74.44	74.52

Table 3.20: Experimental results for $Reddit$ dataset (**bold** are best scores)

which is very different from our setups, where we use larger comments (up to 7 sentences). We also conducted experiments with word and sentence-level attentions (i.e., LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$). Even though we obtain slightly lower accuracy (i.e., 76.67% for the ironic category) in comparison to sentence level attention models, the difference is not as high as other corpus, which we believe is due to the larger size of the training data.

Impact of Size and Nature of Corpus Overall, while the results on the $Reddit$ dataset are slightly better than on the IAC_{v2}, given that the $Reddit$ corpus is ten times larger, we believe that the self-labeled nature of the $Reddit$ dataset might make the problem harder. To verify this hypothesis to this end we conducted two separate experiments. First, we selected a subset of the $Reddit$ corpus that is equivalent to the IAC_{v2} corpus size (i.e., 5,000 examples balanced between ironic and non-ironic category). We utilize the best LSTM model (i.e., attention on prior and current turn) and the model achieves respectively 69.17% and 71.54% F1 for the ironic and non non-ironic category which is lower than what we observe for the IAC_{v2} corpus using the same amount of training data and much lower than the performances reported on Table 3.20. Second, we tried an experiment where we trained our best models (LSTM models with sentence-level attention) on the $Reddit$ corpus and tested on the test portion of the IAC_{v2} corpus. The results, shown in Table 3.21, are much lower than when training using 10 times less amount of data from IAC_{v2} corpus, particularly for the ironic class (more than 10% F1 measure drop). Moreover, unlike all other experiments, adding context does not assist the classifier which seems to highlight a difference between the

Experiment	I			NI		
	P	R	F1	P	R	F1
LSTM$^{ct_{a_s}}$	66.51	61.11	63.69	64.03	69.23	66.53
LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$	63.96	60.68	62.28	62.60	65.81	64.17

Table 3.21: Experimental results for training on $Reddit$ dataset and testing on IAC_{v2} using the best LSTM models (sentence-level attention).

nature of the two datasets including the gold annotations (self-labeled for $Reddit$ vs. crowdsource labeled for IAC_{v2}). We believe the lower performance is due to two main reasons. First, unlike $Reddit$, which is self-labeled, IAC is annotated via crowdsourcing so the nature of the annotations is different between the two corpus. Second, we observe, IAC is predominately based on contentious topics whereas $Reddit$ corpus contains more general topics (i.e., video-games, sports, etc.).

3.6.2.2 Prior Turn and Subsequent Turn as Conversation Context

We also experiment using both the prior turn pt and the succeeding turn st as conversation context. Table 3.22 shows the experiments on the IAC^{+}_{v2} corpus. We observe that the performance of the LSTM models is high in general (i.e., F1 scores in between 78-84$\%$, consistently for both the ironic (I) and non-ironic (NI) classes) compared to the discrete feature based models (i.e.,SVM$_{bl}$). Table 3.22 shows that when we use conversation context, particularly the prior turn pt or the prior turn and the succeeding turn together, the performance improves (i.e., around 3$\%$ F1 for ironic category and almost 6$\%$ F1 improvement for non-ironic category). For the I category, the highest F1 is achieved by the LSTM^{ct}+LSTM^{pt} model (i.e, 83.92$\%$) whereas the LSTM^{ct}+LSTM^{pt}+LSTMst model performs best for the non-sarcastic class (83.09$\%$). In comparison to the attention-based models, although using attention over prior turn pt and successive turn st helps in verbal irony identification compared to the attention over only the current turn ct (i.e., improvement of around 2$\%$ F1 for ironic as well as non-ironic class), generally the accuracy is slightly lower than the models without attention. We suspect this is due to the small size of the IAC^{+}_{v2} corpus ($<$ 3000 instances).

We also observe that the numbers obtained for $IAC{+}_{v2}$ are higher than the one for the IAC_{v2} corpus even if less training data is used. To understand the the difference, we

Experiment	I			NI		
	P	R	F1	P	R	F1
SVM$_{bl}^{ct}$	76.97	78.67	77.81	78.83	77.14	77.97
SVM$_{bl}^{ct+pt}$	76.69	75.0	75.83	76.22	77.85	77.03
SVM$_{bl}^{ct+st}$	67.36	71.32	69.28	70.45	66.43	68.38
SVM$_{bl}^{ct+pt+st}$	74.02	69.12	71.48	71.81	76.43	74.05
LSTM^{ct}	74.84	87.50	80.68	85.47	71.43	77.82
LSTM^{ct}+LSTM^{pt}	80.00	**88.24**	**83.92**	**87.30**	78.57	**82.71**
LSTM^{ct}+LSTMst	79.73	86.76	83.10	85.94	78.57	82.09
LSTM^{ct}+LSTM^{pt}+LSTMst	**81.25**	86.03	**83.57**	85.61	80.71	**83.09**
LSTM$^{conditional(pt->ct)}$	79.26	78.68	78.97	79.43	80.00	79.71
LSTM$^{conditional(ct->st)}$	70.89	69.85	70.37	71.13	72.14	71.63
LSTM$^{ct_{a_s}}$	77.18	84.56	80.70	83.46	75.71	79.40
LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$	**80.14**	83.09	81.59	82.96	**80.00**	81.45
LSTM$^{ct_{a_s}}$+LSTM$^{st_{a_s}}$	75.78	**89.71**	82.15	**87.83**	72.14	79.22
LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$+LSTM$^{st_{a_s}}$	76.58	**88.97**	82.31	87.29	73.57	79.84
LSTM$^{ct_{a_{w+s}}}$+LSTM$^{pt_{a_{w+s}}}$	79.00	80.14	79.56	80.43	79.29	79.86

Table 3.22: Experimental results for $IAC_{v2_{st}}$ dataset using post prior and succeeding turns as context (**bold** are best scores)

analyze the type of the ironic and non-ironic posts from the IAC_{v2}^{+} and found that almost 94% of the corpus consists of ironic messages of "general" type, 5% of "rhetorical questions " type and very few (0.6%) examples of the "hyperbolic" type (Oraby et al., 2016a). Looking at Oraby et al. (2016a) it seems the "general" type obtain the best results (Table 7 in (Oraby et al., 2016a)) with almost 10% F1 over the hyperbolic type set. As we stated before, although IAC_{v2} corpus is larger than the IAC_{v2}^{+} corpus, IAC_{v2} maintains exactly the same distribution of general, rhetorical questions, and hyperbolic examples. This also explains why Table 3.22 shows superior results since classifying the generic category of verbal irony could be an easier task.

3.6.3 Qualitative Analysis

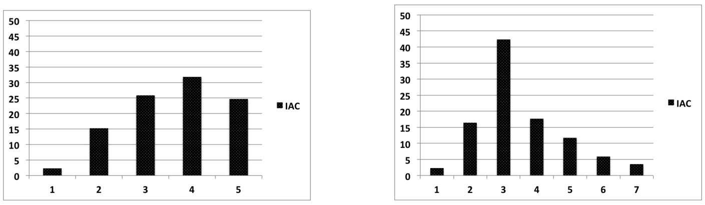
We address the remaining two research questions (e.g., RQ7 and RQ8) in this section. Wallace et al. (2014) showed that by providing additional conversation context humans are able to identify ironic utterances which they were unable to do without the context. However, it will be useful to understand whether a specific *part of the conversation context triggers* the ironic 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 ironic reply and (b) the attention weights of the LSTM models are able to signal similar information. For (a) we designed a crowdsourcing experiment (Crowdsourcing Experiment 1 in Section 3.6.3.1) and for (b) we looked at the attention weights of the LSTM networks (Section 3.6.3.2). This addresses RQ7.

Discussion forums posts are usually long (several sentences), and we noticed in our analysis that computational models have a harder time to correctly label them as ironic or not. Thus, we want to investigate whether there is a particular sentence in the ironic post that expresses the speaker's ironic intent (i.e., addressing the RQ8). To begin to address this issue we conducted another qualitative study to understand whether (a) human annotators are able to identify a sentence in the ironic post that particularly expresses speaker's ironic intent and (b) the sentence-level attention weights are able to signal similar information. For (a) we designed a crowdsourcing experiment (Crowdsourcing Experiment 2 in Section 3.6.3.1) and for (b) we looked at the attention weights of the LSTM networks (Section 3.6.3.2).

For both studies, we compare human annotators' selections with attention weights to examine whether attention weight of the LSTM networks are correlated to human annotations.

3.6.3.1 Crowdsourcing Experiments

3.6.3.1.1 Crowdsourcing Experiment 1 We designed an Amazon Mechanical Turk task (for brevity, MTurk) framed as follows: given an ironic current turn (C_TURN) and its prior turn (P_TURN), we ask Turkers to identify one or more sentences in P_TURN that they think triggered the ironic reply. Turkers could select one or more sentences from the conversation context P_TURN, including the entire turn. We selected all ironic examples from the IAC_{v2} test set where the prior turn, since for longer turns might have been a more complex task for the Turkers. This resulted in 85 pairs. We provided multiple definitions of verbal irony. The first definition is inspired by the Standard Pragmatic Model (Grice et al., 1975) that says verbal irony is a speech or form of writing which means the opposite of what it seems to say. In another definition, taken from Oraby et al. (2016a), we mentioned that verbal irony often is used with the intention to



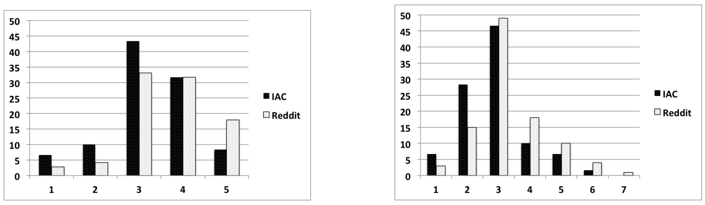
(a)

(b)

Figure 3.6: Crowdsourcing Experiment 1: (a) number of different trigger selections made by the five turkers (1 means all Turkers selected the exact same trigger(s)) and (b) distribution of the number of sentences chosen by the Turkers as triggers in a given post; both in %

mock or insult someone or to be funny. We provided a couple of examples of verbal irony from the IAC_{v2} dataset to show how to successfully complete the task. Each HIT contains only one pair of C_TURN and P_TURN and five Turkers were allowed to attempt each HIT. Turkers with reasonable quality (i.e., more than 95% of acceptance rate with experience of over 8,000 HITs) were selected and paid seven cents per task. Since Turkers were asked to select one or multiple sentences from the prior turn, standard inter-annotator agreement (IAA) metrics are not applicable. To understand the user annotation behaviour though, we look at two aspects. First, we look at the distribution of trigger selection by the five annotators (Figure 3.6. It can be seen that in 3% of cases all five annotators selected the exact same trigger(s), while in 58% of cases 3 or 4 different selections were made per posts. Second, we looked the distribution of the number of sentences in the P_TURN that were selected as triggers by Turkers, and we observe that 43% of time 3 sentences were selected.

3.6.3.1.2 Crowdsourcing Experiment 2 The second study is an extension of the first study. Given a pair of a ironic turn C_TURN and its prior turn P_TURN, we ask Turkers perform two subtasks. First, they were asked to identify "only one" sentence from C_TURN that expresses the speaker's ironic intent. Next, based on the selected ironic sentence, they were asked to identify one or more sentences in P_TURN that may trigger that ironic sentence (similarly to study 1). We select examples both from the IAC_{v2} corpus (60 pairs) as well as the $Reddit$ corpus (100 pairs). Each of the P_TURN and C_TURN contain three to seven sentences (note that the examples from the



(a) (b)

Figure 3.7: Crowdsourcing Experiment 2: (a) number of different trigger selections made by the five turkers (1 means all Turkers selected the exact same trigger(s)) and (b) distribution of the number of sentences chosen by the Turkers as triggers in a given post; both in %

IAC_{v2} corpus is a subset of the ones used in the previous experiment). We replicate the same design as the previous MTurk (i.e, we included definition of verbal irony, gave example, use one pair per HIT, same qualification for Turkers, same payment). Each HIT was done by five Turkers (a total of 160 HITS). To measure the IAA between the Turkers for the first subtask (i.e., identifying a particular sentence from C_TURN that expresses speaker's ironic intent) we used Krippendorf's α (Krippendorff, 2012). We are measuring IAA on nominal data, i.e., each sentence is treated as a separate category. Since the number of sentences (i.e., categories) can vary between three to seven we report separate α scores based on the number of sentences. For C_TURN that contains three, four, five or more than five sentences, the α scores are 0.66, 0.71, 0.65, 0.72, respectively. The α scores are modest and illustrate (a) identifying ironic sentence from a discussion forum post is a hard task and (b) it is plausible that the current turn (C_TURN) contains multiple ironic sentences. For the second subtask, we carried a similar analysis as for experiment 1, and results are shown in Figure 3.7 both for the IAC_{v2} and $Reddit$ data.

3.6.3.2 Comparing Turkers' Answers with Attention Models

In this section we compared the Turker's answers for both tasks with the sentence the sentence-level attention weights of the LSTM models.

To address the RQ7, i.e., the issue of identifying what part of the prior turn triggers the ironic reply, we first measure the overlap of Turkers choice with the sentence-level attention weights of the LSTM$^{ct_{as}}$+LSTM$^{pt_{as}}$ model. For Crowdsourcing Experiment

1, we used the models train/tested on the IAC_{v2} corpus. 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. Figures 3.8 and 3.9 shows side by side the heat maps of the attention weights of LSTM models (LHS) and Turkers' choices when picking up sentences from the prior turn that they thought triggered the ironic reply (RHS). For Crowdsourcing Experiment 2, respectively 51% and 30% of times the sentence with the highest attention weight is also the one picked by Turker for respectively IAC_{v2} an $Reddit$.

To address the last two RQs, i.e., the issue of identifying what sentence of the ironic current turn expresses best the speaker's ironic intent, we again measure the overlap of Turkers choice with the sentence-level attention weights of LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$ model (looking at the sentence-level attention weights from the current turn). We selected the sentence with highest attention weight and matched it to the sentence selected by Turkers using majority voting. For IAC_{v2}, we found that 25% of times the sentence with the highest attention weight is also the one picked by Turkers. For $Reddit$ 13% of times the sentence with the highest attention weight is also the one picked by Turkers. The low agreement on $Reddit$ illustrates that many posts may contain multiple ironic sentences.

For both of these research questions (e.g., RQ7 and RQ8), the obvious question that we need to answer is why these sentences are selected by the models (and humans). In the next section, we conduct a qualitative analysis to try answering this question.

3.6.3.3 Interpretation of Turkers' Answers and Attention Models

We visualize and compare the sentence-level as well as the word-level attention weights of the LSTM models with the Turkers' annotations.

3.6.3.3.1 Semantic Coherence between Prior Turn and Current Turn Figure 3.8 shows a case where the prior turn contains three sentences and the sentence-level attention weights are similar to the Turkers' choice of what sentence(s) triggered the ironic

S1	Ok...
S2	I have to stop to take issue with something here, that I see all too often.
S3	And I've held my tongue on this as long as I can

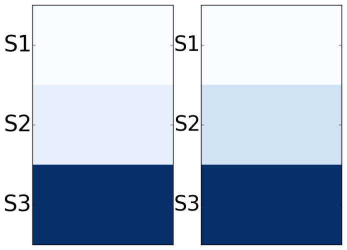


Figure 3.8: Sentences in P_TURN; heatmap of the *attention weights* (LHS) and *Turkers' selection* (RHS) of which of those sentences trigger the ironic C_TURN="Well, it's not as though you hold your tongue all that often when it serves in support of an anti-gay argument."

S1	How do we rationally explain these creatures existence ... for millions of years?
S2	and if they were the imaginings of bronze age ... we can now recognize from fossil evidence?
S3	and while your at it ... 200 million years without becoming dust?

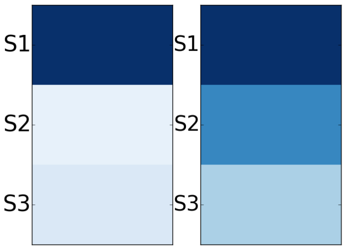


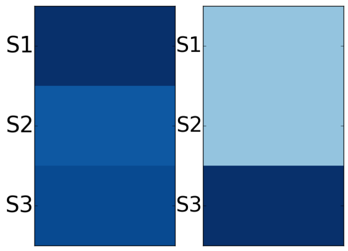
Figure 3.9: Sentences in P_TURN; heatmap of the *attention weights* (LHS) and *Turkers' selection* (RHS) of which of those sentences trigger the ironic C_TURN).

turn. Looking at this example it seems the model pays attention to output vectors that are *semantically coherent* between P_TURN and C_TURN. The ironic C_TURN of this example contains a single sentence – “Well, it’s not as though you hold your tongue all that often when it serves in support of an anti-gay argument”, while the sentence from the prior turn P_TURN that received the highest attention weight is S3 “And I’ve held my tongue on this as long as I can”.

In Figure 3.9, the highest attention weight is given to the most informative sentence –“how do we rationally explain these creatures existence so recently in our human history if they were extinct for millions of years?”. Here, the ironic post C_TURN (userD’s post in Table 2.5) mocks userC’s prior post (“how about this explanation – you’re reading waaaaay too much into your precious bible”). For both figures — Figure 3.8 and Figure 3.9, the sentence from the prior turn P_TURN that received the highest attention weight has also been selected by the majority of the Turkers. For Figure 3.8 the distribution of the attention weights and Turkers’ selections are alike. Both examples are taken from the IAC_{v2} corpus.

Figure 3.10 shows a conversation context (i.e., prior turn) and the ironic turn (userE and userF’s posts in Table 2.5) together with their respective heatmaps that reflect the two subtasks performed in the second crowdsourcing experiment. The bottom part of the figure represents the sentences from the C_TURN and the heatmaps that compares attention weights and the Turkers’ selections for the first subtask: selecting the sentence from C_TURN that best expresses the speaker’s ironic intent. The top part of the figure shows the sentences from the P_TURN as well as the heatmaps to show what sentence(s) are more likely to trigger the ironic reply. We make two observations: (a) Turkers have selected multiple sentences from the C_TURN as expressing verbal irony. The Attention model has given highest weight to the last sentence in C_TURN similar to the Turkers’s choice; (b) The attention weights seem to indicate semantic coherence between the ironic post (i.e, “nothing to see here” with the prior turn “nothing will happen, this is going to die …”). We also observe similar behavior in Tweets (highest attention to words –*majority* and *gerrymadering* in Figure 3.12).

P_TURN	S1	nothing will happen, . . . other private member motions.
	S2	this whole thing is being made . . . are trying to push an agenda.
	S3	feel free to let your . . . discussion before it is send to the trashcan.



C_TURN	S1	the usual "nothing to see here" response.
	S2	whew!
	S3	we can sleep at night and ignore this.

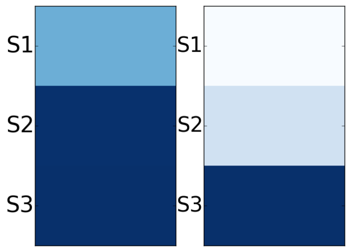


Figure 3.10: Sentences from P_TURN that trigger verbal irony (Top) and Sentences from C_TURN that express verbal irony (Bottom). Tables show respectively the text from P_TURN and C_TURN (top and bottom) and Figure shows the heatmap of *attention weights* (LHS) and *Turkers' selection* (RHS)

S1	you disguting sickening woman!
S2	how can you tell a man that about his mum??!!!
S3	evil twisted filthy...

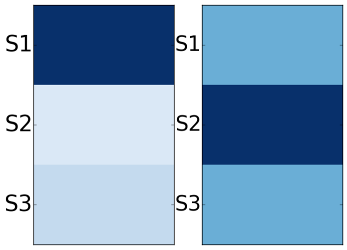


Figure 3.11: Sentence from P_TURN and the heatmaps of the *attention weights* (LHS) and *Turkers' selection* (RHS) emphasizing which of these sentences trigger the ironic C_TURN

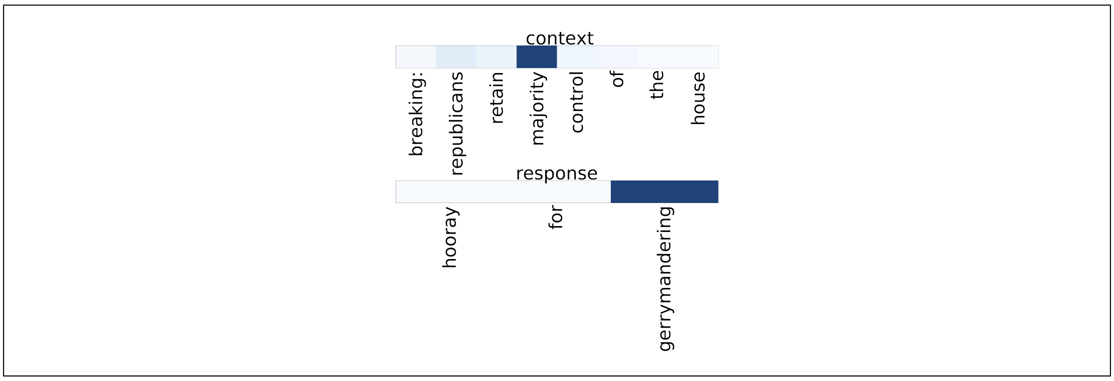


Figure 3.12: Attention visualization of semantic coherence between c and r

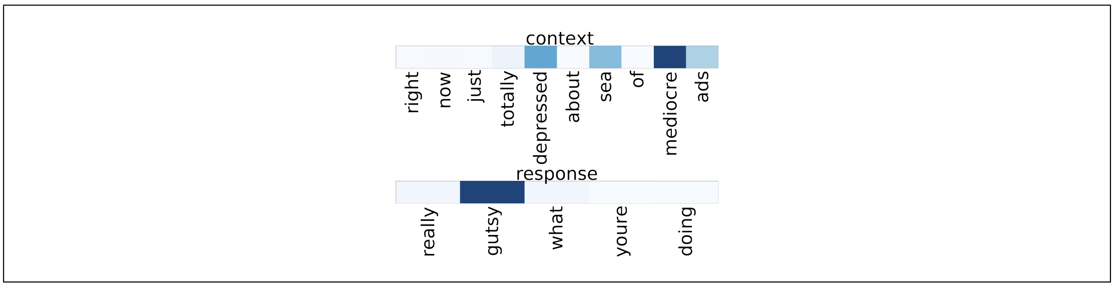


Figure 3.13: Attention visualization of incongruity between P_TURN and C_TURN

3.6.3.3.2 Attention Weights and Irony Markers We observe that attention weights are also correlated to the different aspects of irony (e.g., irony markers and irony factors). For example, looking just at attention weights in reply, we notice the models are giving highest weight to sentences that contain irony markers, such as emoticons (e.g.,, ":p", ":)") and interjections (e.g., "ah", "hmm"). We also observe interjection such as "whew" with exclamation mark receive high attention weight (Figure 3.10; see the attention heatmap for the current turn C_TURN).

3.6.3.3.3 Incongruity between Conversation Context (P_TURN) and Current Turn (C_TURN) Finally, we observe that attention weights sometime are correlated to the semantic incongruence of irony. It is possible that the literal meaning of the current turn C_TURN is incongruent with the conversation context (P_TURN). We observe in discussion forums and in Tweets that the attention-based models have frequently identified sentences and words from P_TURN and C_TURN that are semantically incongruous. For instance, in Figure 3.11, the attention model has chosen sentence S1, which contains

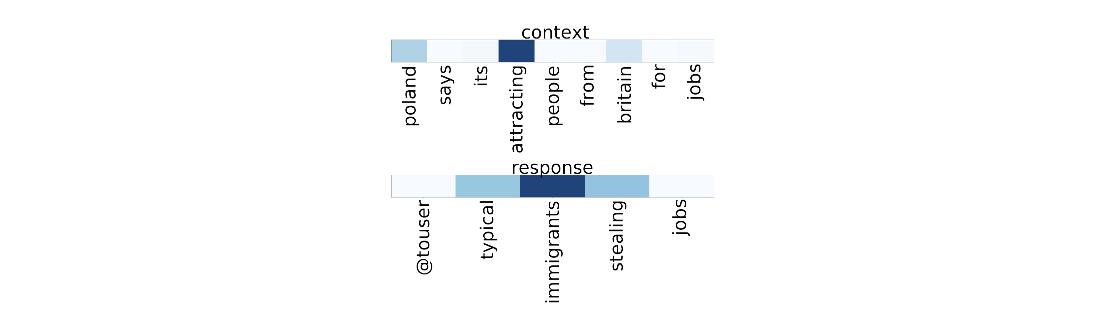


Figure 3.14: Attention visualization of incongruity between P_TURN and C_TURN

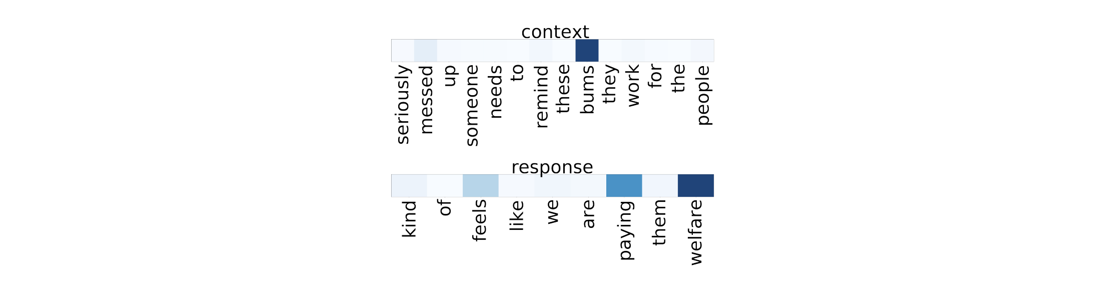
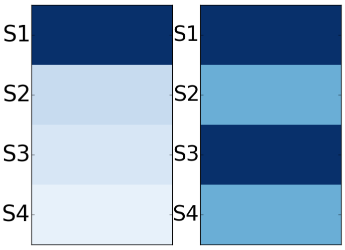


Figure 3.15: Attention visualization of incongruity between P_TURN and C_TURN

strong negative sentiment words ("you disgusting sickening woman"). In contrast, the attention model on the current turn C_TURN, has given the highest weight to sentence that contain opposite sentiment ("I love you"). Thus, the model seems to learn the incongruity between the prior turn P_TURN and the current turn C_TURN in terms of the opposite sentiment. However, from Figure 3.11, it seems the Turkers prefer the second sentence S2 ("how can you tell a man that about his mum?") as the most instructive sentence instead of the first sentence. Figure 3.16 shows a pair of prior turn P_TURN and the current turn C_TURN where the attention model has picked up the opposite sentiment of "protecting home from a looter" (i.e., current turn) w.r.t to the context of "chose to fight" or "he died because of it" (first and second sentence from P_TURN).

In Twitter dataset, we observe that the attention models often have selected utterance(s) from the context which have opposite sentiment (Figure 3.13, Figure 3.14, and Figure 3.15). Here, the word and sentence-level attention model have chosen the particular utterance from the context (i.e., the top heatmap for the context) and the words with high attention (e.g., "mediocre", "gutsy"). Word-models seem to also work well when

	S1	technically speaking : this guy chose to fight in the ukraine.
P_TURN	S2	he died because of it.
	S3	sure russia fuels the conflict, but he didnt have to go there.
	S4	his choice, his consequences.



	S1	sure thing.
C_TURN	S2	protecting your home from an looter?
	S3	nope, why would anyone do that?

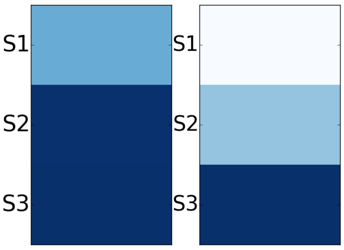


Figure 3.16: Sentences from P_TURN that trigger verbal irony (Top) and Sentences from C_TURN that represents verbal irony (Bottom). Tables show respectively the text from P_TURN and C_TURN (top and bottom) and Figure shows *attention weights* (LHS) and *Turkers' selection* (RHS)

words in the prior turn and current turn are semantically incongruous but not related to sentiment ("bums" and "welfare" in context: "someone needs to remind these *bums* they work for the people" and reply: "feels like we are paying them *welfare*" (Figure 3.15).

3.7 Conclusion

There are three main contributions of this chapter. First, we provided a thorough analysis of two specific aspects of verbal irony – irony markers and irony factors. We analyzed irony markers from different platforms, $Twitter$ and discussion forum such as $Reddit$. We provided a detailed statistical analysis of several irony markers and showed that how different markers (e.g., emoticons) are more common to platform such as $Twitter$ whereas, for discussion forums users are more keen to use hyperbolic words or metaphors. We analyzed two particular irony factors, reversal of valence and incongruence in irony. We reframe the irony detection problem as a word-sense disambiguation problem and proposed a new SVM Kernel based on word embeddings that achieve a very high accuracy. We also model incongruence in irony and showed that the target centric-CNN model (two CNN model) achieves the best accuracy in identifying ironic utterances based on incongruence.

Second, we investigated verbal irony with and without any contextual information. We utilize conversation context if an irony appears in a dialogue and showed how such context assists in achieving higher accuracy. We implemented a novel architecture based on the LSTM networks where each turn is read by a separate LSTM and showed how attention-based LSTM network performs best when context is added in the classification. We also provided a detailed user study to investigate two questions: (1) given an ironic message that contains multiple sentences, can humans and computational models identify the specific sentence that is ironic (2) can humans and computational models identify what part of the prior turn (i.e., conversation context) triggered the ironic post.

Finally, the third contribution of this chapter is the different resources that we have released for research on verbal irony identification. We analyzed both self-labeled (e.g., tweets and $Reddit$ posts) as well as crowdsourced (e.g., AC corpus) datasets.

Chapter 4
Verbal Irony Interpretation

4.1 Overview

Verbal irony or sarcasm is a type of interactional phenomenon with specific perlocutionary effects on the hearer (Haverkate, 1990).[1] Verbal irony is always intended for an audience, and thus we argue that besides *locating and identifying ironic utterances in natural discourse such as social media*, it is equally important to understand *how the audience interprets* verbal irony. Such interpretation of ironic utterances largely depends upon the shared knowledge of speaker/author and hearer about the situation of the ironic utterance (Haverkate, 1990). Most computational approaches for sarcasm or verbal irony have focused on detecting whether a speaker is ironic or not (Davidov et al., 2010; González-Ibáñez et al., 2011; Riloff et al., 2013; Liebrecht et al., 2013; Maynard and Greenwood, 2014; Joshi et al., 2015; Muresan et al., 2016; Ghosh and Veale, 2016; Wallace et al., 2014; Amir et al., 2016; Bamman and Smith, 2015; Schifanella et al., 2016). In contrast, there is little computational research on how hearers interpret ironic utterances and what are the strategies that they use for irony interpretation.[2] This gap motivates our research in the current chapter.

This chapter aims to deepen our understanding of how the hearers interpret verbal irony by introducing a typology of linguistic strategies. We leverage the crowdsourcing task introduced in the previous chapter for detecting the literal vs. ironic sense of the utterances (Section 3.5.2.1). The task was framed as follows: given a speaker's ironic message, five Turkers on Amazon Mechanical Turk (MTurk) are asked to re-phrase the

[1] Part of the chapter is now under review.

[2] Hearers and readers are used interchangeably in this chapter.

message to express the speaker's intended meaning. These re-phrasings are verbalizations of how the hearers' interpret the ironic message (see Table 4.1; I$_{im}$ denotes the speaker's ironic message, while H$_{int}$ denotes the hearer/Turker's interpretation of that ironic message). Now consider the third example from the Table 4.1. The ironic message I$_{im}$ "pictures of you holding dead animal carcasses are so flattering" is rephrased by three Turkers. For this utterance, the strength of negative sentiment perceived by the hearer depends on whether they interpret the speaker's actual meaning as "picture ... are **not flattering**" vs. "pictures ... are **so gross**...". Here, the intensity of negative sentiment is higher in the latter interpretation than in the former. In this chapter, we propose a typology of linguistic strategies to categorize such verbalization of hearers' interpretations. Our main motif is to analyze different linguistic strategies used to interpret irony. We also provide empirical methods to identify the strategies automatically.

I$_{im}$	H$^1_{int}$	H$^2_{int}$	H$^3_{int}$
1. loved hearing that read out, so pleased your all taking notice	I don't want to hear that read out	hated hearing that read out, so unhappy your all taking notice	can't say I enjoyed that read out of your notice.
2. can't believe how much captain America looks like me	I wish I looked like Captain America. I need to lose weights	can't believe how much captain America looks different from me	I don't, but I wish I looked like Captain America
3. Pictures of you holding dead animal carcasses are so flattering	Hate hunting season and the pictures of you holding dead animal are so gross	Pictures of you holding dead animal carcasses is an unflattering look	Pictures of you holding dead animal carcasses are not flattering
4. I also believe everything i read . the media here is totally unbiased	I think the media is very biased	I never believe everything i read. the media here is completely biased, on all sides	I do not believe everything i hear
5. AWWW you still get on myspace ! How cute !	Myspace is not cool anymore.	Myspace is so outdated and nobody but you uses it.	It is very behind the times to use myspace

Table 4.1: Examples of speaker's ironic messages (I$_{im}$) and interpretations given by 3 hearers/Turkers (H$^i_{int}$).

This chapter makes three contributions. Although studies in linguistics as well as in psychology analyze readers/hearers' interpretation, they mostly examine the *processing time* of literal vs. non-literal interpretations (Giora et al., 1998; Dews and Winner, 1999; Ivanko and Pexman, 2003). Unlike these approaches, our contribution is a data-driven *typology of linguistic strategies* that hearers use to interpret ironic messages. (Section 4.4). Second, we propose computational models to capture these strategies and perform a comparative analysis of their distribution on two different datasets (Section 4.5). Third, we present two user studies that aim to answer two questions: (1) do interpretation strategies of verbal irony vary by hearer? and (2) how do expressions of verbal irony influence the choice of interpretation strategy by hearers? (Sections 4.6 and 4.7).

Before we present our research, we describe related research in the next section.

4.2 Related Work and Background

Although there is agreement in theoretical linguistics in conceiving verbal irony or sarcasm as a mismatch between what the speaker says and its intended meaning, different explanatory accounts of verbal irony have been proposed leading to a wide range of theories such as (Neo) Gricean Theories (Grice, 1978), Mention theory (Wilson, 2006; Wilson and Sperber, 2002), Pretense Theory (Clark and Gerrig, 1984), Direct Access Model (Gibbs Jr, 2003), the Indirect Negation Theory (Giora, 1995) and the Display Theory (Williams, 1983), to cite a few. In the previous chapter we introduced these competing theories of irony analysis in detail. We observe that the majority of the debates on irony analysis are based on whether users process literal and ironic utterances in the same way or are there differences in processing between ironic and literal meaning. Many scholars have focused on measuring the processing time of ironic and literal messages and to determine the effect of contextual knowledge in processing ironic utterances. Another much debated issue is the nature of the verbal irony phenomena: whether it is a binary rather than a gradual phenomenon, are utterances either ironic or non-ironic or are there different degrees of irony at the speakers disposal (Giora, 1995; Gibbs Jr, 2003)? Here we discuss some of the theoretically motivated empirical research that examine the processing mechanism of irony.

The predictions of the Standard Pragmatic Model (for brevity, SPM (Grice, 1978)) have been tested by a number of behavioral studies that compare the reading time between ironic and literal utterances. Recall, SPM predicts that readers of an ironic message first process the literal assessment of an utterance and when they find the literal evaluation is incongruent with the context, the intended (i.e., ironic) evaluation is activated. Dews and Winner (1999) measured reading times in response to discourse reading where the last sentence was either ironic or non-ironic. They found longer reaction times were recorded for ironic utterances compared to the equivalent non-ironic meaning. A later study by Schwoebel et al. (2000) also supported the SPM and argued that some aspects of the literal meaning always have to be processed during the interpretation of verbal irony that results in longer time for processing ironic utterances.

In contrary to the SPM, the "direct access model" proposed by Gibbs (1986) claims that speakers do not have to understand the entire literal meaning of the linguistic expression before accessing the ironic meaning. Rather, the interpretation of ironic meaning largely depends upon the pragmatic knowledge and figurative modes of thought of the listener. Behavioral studies exist that show identical reaction times for the comprehension of ironic and non-ironic messages (Gibbs Jr et al., 1995). However, Regel (2009) observed that reading times were measured based on the comprehension of full utterance without measuring the processing of any particular terms (e.g., that may trigger irony). Regel (2009) also argued that it is possible that processing of ironic or sarcastic utterances needs extra or different processing mechanism, however, that might not effect the final outcome (i.e., comprehension time of the full sentence).

Based on the hypotheses of "direct access model" Ivanko and Pexman (2003) studied the effect of the contextual information by manipulating the degree of situational negativity (using (a) strongly negative, (b) weakly negative, and (c) neutral contexts). They found that participants took more time to read ironic statements that followed strongly negative contexts. In contrary, for *weakly negative situations*, (i.e., case (b)), reading times for ironic statements were faster than or equivalent to reading times for literal utterances. This is somewhat similar to the findings of (Utsumi, 2000) who did not agree that literal language is always processed faster than irony. Instead, Utsumi (2000) mentions the use of the prototypical irony (e.g., verbal irony that involves the hearer having an expectation, this expectation not being met, and the presence of a negative emotional attitude toward the incongruity between expectation and outcome (Ivanko and Pexman, 2003)) are processed in the same time as the same statement uttered literally.

The research described in Ivanko and Pexman (2003) is also related to the previous study on understanding the effect of the situational disparity (i.e., semantic incongruence) in irony processing. For instance, Gerrig and Goldvarg (2000) examined the influence of the degree of situational disparity on the perception of irony and found that greater situational variation can lead to a higher perception of irony. Likewise, Colston and O'Brien (2000) manipulated the degree of contrast between a context situation and a ironic statement by varying the degree of sentiment word (e.g., strong vs. weak. vs. neutral sentiment word in a ironic utterance). They found when there was a high degree of difference between the strong and weak version of statements, the speakers of strongly ironic utterances were rated to be more condemning and more humorous than the speakers of weakly ironic statements.

In contrary to the "direct access model", Giora (1995, 1997) propose an alternative view of irony processing. This is named as the Graded Salience Hypothesis (GSH), which states that *salient* (e.g., conventional, frequent, familiar) meanings should be activated before the less salient meanings. Thus, the differences in reaction times for figurative and literal sentences have been suggested to result from differences in salience of meanings (Giora and Fein, 1999). The authors argued that processing of conventional form of irony (also denoted as "stock irony"; see Burgers et al. (2012)) takes identical time as their literal interpretation readers/hearers are familiar with such instances. In contrary, processing of unconventional form of irony needs longer processing time.

Scholars have also identified various contextual features, characteristics such as the gender and occupation to understand the speakers and addresses of the ironic utterances (Burgers, 2010; Katz and Pexman, 1997; Gibbs and Izett, 2005). Gibbs and Izett (2005) have categorized the addressees into two groups; a group of people who understand irony and the group of people who do not.

4.3 Datasets of Speaker's Ironic Messages and Hearer's Interpretations

In the last chapter we discussed the crowdsourcing experiment to identify words that may have literal or ironic meaning. Given a speaker's ironic message (I_{im}), five Turkers (hearers) on MTurk are asked to re-phrase the message so that the new message is likely to express the speaker's intended meaning (H_{int}). This generated a parallel dataset (i.e., I_{im} - H_{int}) and we applied monolingual alignment (Barzilay and McKeown, 2001) to extract words for the reversal of valence experiment (Section 3.5.2.1). We use the same dataset now to study the verbalization of hearers' interpretation to analyze what the speaker actually meant to say.

Peled and Reichart (2017) employed similar design to generate a parallel dataset to use for generating interpretations of ironic messages using machine translation approaches. Instead of using novice annotators (Turkers), they use workers skilled in comedy writing and literature. Since the paraphrases (we refer to (Peled and Reichart, 2017)'s dataset as $SIGN$) also could be used as parallel dataset similar to I_{im} - H_{int}, we use the $SIGN$ dataset too in our experiments. However, there are two major differences between our dataset I_{im} - H_{int} and the $SIGN$ dataset. First, we focus on verbal irony, and we always require an interpretation from the workers. Peled and Reichart (2017) indicated that they did not ask the annotators to always rephrase the utterances (i.e., so all the rephrases are not intended meaning). Second, to carry out the hearer-dependent study we need information about the users who performed the interpretation task, which is not readily available in the $SIGN$ datasets.

Here, we explain the main characteristics of the MTurk experiment that generated the parallel dataset of I_{im} - H_{int}. We collect our data from Twitter since we focus on verbal irony: the use of hashtags gives us gold labels that point to the speakers' intent of being ironic. We collected 1,000 English utterances using the hashtags #ironic and #sarcasm. [3] The Turkers were presented with detailed instructions of the task including definition of verbal irony, the task description, and multiple examples. We provided the

[3] We manually checked the utterances to remove any tweets that are not clearly ironic.

standard definition of verbal irony from the Merriam Webster dictionary (i.e., verbal irony conveys a meaning that is the opposite of the literal meaning.).

The Turkers were also instructed to consider the entire message in their rephrasings. This emphasis was added to avoid high asymmetry in length between the speaker's ironic message (I_{im}) and the rephrasing that express the hearer's interpretation (H_{int}). We also asked the Turkers to rate the difficulty of the task with a score between one and five, where one means very easy and five means very difficult. After the rephrasing task is completed, we obtained a dataset of 5,000 I_{im}-H_{int} pairs. Table 4.1 shows examples of speaker's ironic messages (I_{im}) and their corresponding hearers' interpretations. Each HIT contained one ironic message and the Turkers were paid five cents for each HIT. Next, we ran a second MTurk task to verify whether the generated H_{int} messages are correct re-phrasings of the ironic messages. We included qualification tests so that only those Turkers who were not involved in the content generation task were allowed to perform this task. For each task three Turkers were allowed to verify the rephrasing with one of the three options (e.g., "yes" (i.e., correct rephrasing), "no" (i.e., wrong rephrasing), "I cannot decide"). Turkers were paid five cents for each HIT. We applied majority voting, i.e., accepting only those rephrasing that have received at least two "yes" votes from the Turkers. They labeled 5% of H_{int}s (i.e., 238 rephrasing) as invalid and low quality. This set of wrong H_{int}s include empty strings, H_{int}s with only few words, complete copies of the original messages, and wrong interpretation of a ironic message. To assure a first level of quality control, for both MTurk tasks we allowed only qualified Turkers (i.e., at least 95% approval rate and 5,000 approved HITs).

4.4 Interpreting Verbal Irony: A Typology of Strategies

In this section we propose a typology of linguistic strategies used in hearers' interpretations of speakers' ironic messages. Given the definition of verbal irony, we would expect that Turkers' interpretation of speaker's intended meaning will contain some degree of opposite meaning w.r.t. to what the speaker's said. However, it is unclear what linguistic strategies the Turkers will use to express that. Our methodological assumption is that since verbal irony is an interactional phenomenon the analysis of how ironic utterances are interpreted by potential hearers is crucial to identify linguistic strategies expressing irony.

To build our typology, from the total set of I$_{im}$-H$_{int}$ pairs obtained through crowdsourcing (e.g., 4,762 pairs; see Section 4.3) we selected a dev set of 500 I$_{im}$-H$_{int}$ pairs to manually compare and contrast each pair of utterances containing speaker's ironic message and hearer's interpretation. This dev set provides a testbed for the explanatory potential of theoretical frameworks. In the following subsection we discuss the identified linguistic strategies, and then link these strategies to theoretical framework of verbal irony.

4.4.1 Linguistic Strategies

Lexical and phrasal antonyms: This category refers to lexical antonyms (e.g., "love" ↔ "hate", "great" ↔ "terrible"). We have also considered antonyms which, despite not being the direct antonym of the corresponding term in the ironic sentence, bear an opposite meaning in the context of utterance, i.e., "ed davey is such a *passionate*, inspiring speaker" →"ed davey is such a *boring*, inspiring speaker", working as lexical opposite. Although typical antonyms of "passionate" are "unpassionate" and "uncaring", "boring" works in this context as a lexical *opposite* since the fact that a speaker is passionate entails that he is not boring. We label this set of antonyms as *indirect* antonyms. This strategy, by numbers is the most popular strategy taken by the Turkers during interpretation. In other words, Turkers perceived the "opposite meaning" of the ironic utterance and used a lexical antonym in the rephrasing. 42.2% of times this strategy is used in the dev set.

Besides lexical antonyms, Turkers sometimes use antonym phrases (e.g., "I can't wait" → "not looking forward", "I don't like" → "I am upset"). Antonym phrases were used in 5% of cases. However, as we implement empirical models to identify the strategies we observe that the lexical choices used to build the phrases are difficult to predict (Section 4.5.1). The strategy of antonyms (direct/indirect/phrases) is used by Turkers mainly when the trigger of irony in the ironic utterance is a word/phrase expressing sentiment.

Negation: Turkers change the polarity of the ironic sentence negating the main predicate. This strategy is used in the presence of copulative constructions where the predicative expression is an adjective/noun expressing sentiment (e.g., “is great” → “is **not** great”) and of verbs expressing sentiment (e.g., “love” → “**no** love”) or propositional attitudes (e.g., “I wonder” → “I **don’t** wonder”). Recall in Table 4.1 the ironic message “. . . dead animal carcasses are so flattering” is rephrased “picture . . . are **not flattering**” by a Turker. Thus, the intensity of the negative sentiment is lower in this interpretation. 28.4% of times this strategy is employed in the dev dataset.

Weakening the intensity of sentiment: The use of negation and antonyms is sometimes accompanied by two strategies that reflect a weakening of sentiment intensity. First, when I$_{im}$ contains words expressing high degree of positive sentiment, the hearer’s interpretation replaces them with more neutral ones (e.g., “**love**” → “don’t **like**”). Second, when I$_{im}$ contains an intensifier, it is eliminated in the Turkers’s interpretation. Intensifiers are linguistic items which specify the degree of the value/quality expressed by the words they modify (Méndez-Naya, 2008) (e.g., “cake for breakfast. I am **so** healthy” → “cake for breakfast. I am **not** healthy”). We also observe that intensifiers as well as words expressing a high degree of sentiment express the trope of hyperbole. Therefore, it seems that this figure of speech is perceived as an *irony marker* (Attardo, 2000a), a meta-communicative clue which instructs hearers in the interpretation process: if the linguistic items expressing hyperbole are removed, the sentence is still ironic, but less easy to interpret. This strategy is used 23.2% of times.

Antonym/Negation + Interrogative to Declarative Transformation. Another strategy, used in conjunction with the negation or antonym strategies is to replace the interrogative form with an declarative form, when I$_{im}$ is a rhetorical question (for brevity, RQ) (e.g., (“don’t you **love** fighting?” → “I **hate** fighting”). RQs are deemed as a pragmatic strategy (Muecke, 1969) to be accounted for in terms of pretense and indirect assertion (Kaufer, 1981; Recanati, 2004; Frank, 1990): the speaker does not mean to ask a question but simply pretends to perform one. These strategy accounted for 5% of cases.

Counterfactual Desiderative Construction: Counterfactual desiderative constructions are distinguished from other strategies again from a pragmatic rather than semantic point of view: they point to failed expectations as the origin of irony. When the ironic utterance expresses a positive/negative sentiment towards a past event (e.g. "glad you relayed this news . . . ") or an expressive speech act (e.g. "thanks X that picture needed more copy") the hearer's interpretation of intended meaning is expressed through the counterfactual desiderative construction *I wish (that) p* ("I wish you hadn't relayed . . . ", "**I wish** X didn't copy . . . "). This strategy is used 2.8% of times.

Pragmatic Inference: In addition to the above strategies, there are cases where the interpretation calls for an inferential process to be recognized. For instance, the utterance "made 174 this month . . . *I'm gonna buy a yacht!*" → "made 174 this month . . . **I am so poor**". Pragmatic inference strategy was applied in 3% of cases. The distribution of the strategies is represented in Table 4.2.

4.4.2 Links to Theoretical Frameworks

The linguistic strategies of *antonyms* and *negation* are in line with the Gricean (Grice et al., 1975) account of irony: they show that ironic messages are perceived as messages uttered to convey a meaning opposite to that literally expressed, flouting the conversational maxim of Quality "do not say what you believe to be false". In messages expressing verbal irony, the violation of the maxim is frequently signaled by "the opposite" of what is said literally (e.g., intended meaning of "carcasses are flattering" is they are gross; Table 4.1).

The Gricean account for irony based on pragmatic insincerity does not provide an explanation for the cases where the strategy is **pragmatic inference**, since the speaker is not just uttering a falsehood to mean the opposite (e.g., "I am not going to buy a yacht"). The Relevance Theory account of irony (Wilson, 2006; Wilson and Sperber, 2012) seems more suitable here: the sentences *I'm gonna buy a yacht!* echoes some common ground knowledge (e.g., "to buy a yacht requires a high salary"), which instruct the hearer in inferring the speaker's ironic intent.

Typology	Distribution (%)
Antonyms	
- lexical antonyms	(42.2)
- antonym phrases	(6.0)
Negation	
- simple negation	(28.4)
Antonyms OR Negation	
- weakening sentiment	(23.2)
- interrogative → declarative	(5.2)
- desiderative construction	(2.8)
Pragmatic inference	(3.2)

Table 4.2: Typology of linguistic strategies and their distribution (in %) over the dev set

4.5 Empirical Analysis of Interpretation Strategies

Our goal is to perform a comparative empirical analysis to understand how the hearers use the typology. To accomplish this, we propose computational models to automatically detect these strategies in two datasets: (1) our I_{im} -H_{int} dataset and (2) the $SIGN$ dataset released by Peled and Reichart (2017). As stated earlier in Section 4.3, albeit for a different purpose, their task design is identical to ours: they used a set of 3,000 ironic tweets and collected five rephrasings, including an option not to rephrase, using workers skilled in comedy writing and literature paraphrasing. $SIGN$ contains 14,970 pairs. To evaluate our models, we asked two annotators to annotate two $test$ set of 500 pairs each from I_{im} -H_{int} and the $SIGN$ dataset (i.e., denoted by $SIGN_{test}$). In the 500 pairs in $SIGN_{test}$, 79 contain no rephrasings (the workers just copied the original message), which we eliminated, thus, the $SIGN_{test}$ contains only 421 instances.

4.5.1 Computational Methods

Lexical Antonyms: To detect whether an I_{im}-H_{int} pair uses the *lexical antonym* strategy, we first need to built a resource of lexical antonyms. We use: (a) the MPQA Lexicon (Wilson et al., 2005) of over 8,000 positive, negative, and neutral sentiment words, (b) an opinion lexicon with around 6,800 positive and negative sentiment words (Hu and Liu, 2004b), (c) the list of antonym pairs identified from contrasting adjacent thesaurus categories Mohammad et al. (2013), (d) antonyms from WordNet, and (e) opposite verbs from the VerbOcean (Chklovski and Pantel, 2004).

In addition, we also automatically extract lexical antonyms from the parallel datasets (I$_{im}$-H$_{int}$). Specifically, we use the monolingual alignment approach of Barzilay and McKeown (2001), which has been shown to be biased towards word-level paraphrases (Bannard and Callison-Burch, 2005). From Table 4.1, considering the first I$_{im}$ paired with the H$^{3}_{int}$, this algorithm extracts "loved" and "hated" as paraphrases. This way we can identify antonym pairs such as, "flattering" $\leftrightarrow$ "gross", "brilliant" $\leftrightarrow$ "stupid". Since not all extracted lexical paraphrases are guaranteed to be antonyms, we manually evaluated the extracted list and selected 393 lexical antonym pairs that augment the above lexicons.

After accumulating all the antonym resources we turn into automatically recognizing the antonyms from the I$_{im}$-H$_{int}$ pairs. Given the created lexicons of lexical antonyms, the task is to detect whether a given I$_{im}$-H$_{int}$ pair uses the *lexical antonym* strategy. One baseline is just search whether a lexical antonym pair is present (P/R/F1 scores are respectively 58.6%, 87.9%, and 70.4$\%$ respectively on the dev data). This baseline is quite noisy (low precision) since we are unsure whether the lexical antonyms are in similar syntactic contexts. Thus, we propose an approach that uses word-alignments and dependency parse trees. The dependency trees are acquired via the Stanford NLP parser (De Marneffe et al., 2006), while the word-to-word alignments are extracted using a statistical machine translation (SMT) alignment method - IBM Model 4 with HMM alignment implemented in Giza++ (Och and Ney, 2004). For a I$_{im}$-H$_{int}$ pair, we first select all the candidate lexical antonym pairs using the lexicons introduced above. Next, we look at the word-alignments between I$_{im}$-H$_{int}$ pair to discover whether a lexical antonym pair is aligned (similarly to approaches for contradiction detection in De Marneffe et al. (2008)). This results in high precision but low recall (P/R/F1 scores are 88.6%, 50.5%, and 64.3$\%$ respectively on the dev data). To improve the recall, we then look at the dependency trees of the pair. Below is the short description of the algorithm (line number 6 to 13) that use word-word alignment as well as the dependency trees ($Antonym_{match}$ (**Algorithm 1**)).

For a I$_{im}$-H$_{int}$ pair, we first select all the candidate lexical antonym pairs using the

Algorithm 2 $Antonym_{match}$

```
1: procedure ($I_{im}$, $H_{int}$, $I_{im_{dep}}$, $H_{int_{dep}}$,$align_{I_{im},H_{int}}$)
2:     $a_{dict} \leftarrow antonyms(I_{im})$
3:     for $k \leftarrow 0, a_{dict}.keys()$ do
4:         $I_{im_k} \leftarrow I_{im_{dep}}.get(k)$
5:         $I_{im_{k_m}} \leftarrow I_{im_{dep}}.mod(I_{im_k})$
6:         for $v \leftarrow 0, a_{dict}.get(k)$ do
7:             $H_{int_v} \leftarrow H_{int_{dep}}.get(v)$
8:             $H_{int_{k_v}} \leftarrow H_{int_{dep}}.mod(H_{int_v})$
9:             if $I_{im_{k_m}} = H_{int_{k_v}}$ then
10:                $Strategy_{anto} \leftarrow True$
11:            end if
12:            if $align_{I_{im},H_{int}}(I_{im_k}, H_{int_v}) \leftarrow True$ then
13:                $Strategy_{anto} \leftarrow True$
14:            end if
15:        end for
16:    end for
17: end procedure
```

lexicons introduced above. If the antonym nodes are the roots of the dependency trees then *lexical antonym* strategy is selected. If not, we look at the nodes modified by the lexical antonyms in the respective trees, and if they are the same we select *lexical antonym* strategy [line 9 to 11 in **Algorithm 1**]. Given the pair “can you show any **more** of the steelers” → “. . . **less** of the steelers”, the candidate lexical antonyms are *more* and *less* and they are the objects of the same predicate in I$_{im}$-H$_{int}$: **show**. We also look at the word-alignments between I$_{im}$-H$_{int}$ pair to discover whether the lexical antonyms are aligned [line 12 -13 in **Algorithm 1**]. Out of 211 I$_{im}$-H$_{int}$ pairs that are annotated as having *lexical antonym* strategy (dev set), 12 instances are identified by the dependency parse method, 67 instances by the word-alignment method, and 100 instances by both methods (P/R/F1 scores respectively are 92.1%, 77.7% and 84.3%). However, sometimes both dependency and word-alignment method fails. In Twitter, users often use ellipsis to save the number of characters and words with hashtags. For example, in “ . . . good day circling down the toilet bowl. **Yay**”. → “good day is circling down the toilet bowl . . . **awful**.”, although the lexical antonyms **yay** and **awful** exist, neither the alignment nor the dependency trees method is able to detect it. We found 25 such instances in the dev set. To account for this, once we run the dependency and alignment methods, we also just look whether a I$_{im}$-H$_{int}$ pair contains a lexical antonym pair. On the dev set (Table 4.5) our approach shows 89.0% precision, 95.7% recall and 92.2% F1 measure. This strategy is denoted as Lex_ant Strategy in Table 4.5.

Negation: This interpretation strategy involves identifying the presence of negation and its scope. Here, however, the scope of negation is constrained since generally Turkers negated only a single word (i.e., "**not** love"). Thus, our problem is easier than the general problem of finding the scope of negation (Reitan et al., 2015; Fancellu et al., 2016; Prabhakaran and Boguraev, 2015). Our algorithm of negation detection is depicted in $Negation_{match}$ (**Algorithm 2**).

Algorithm 3 $Negation_{match}$

```
 1: procedure ($I_{im}$, $H_{int}$, $I_{im_{dep}}$, $H_{int_{dep}}$, $align_{I_{im},H_{int}}$)
 2:     $neg_{list} \leftarrow negations()$
 3:     for $neg \leftarrow 0, neg_{list}$ do
 4:         if $neg \in H_{int_{words}}$ then
 5:             $H_{int_{m_{dep}}} \leftarrow H_{int_{dep}}[neg]$
 6:             if $H_{int_{m_{dep}}} \in I_{im_{dep}}$ then
 7:                 $Strategy_{neg} \leftarrow True$
 8:             end if
 9:             $I_{im_{m_{dep}}} \leftarrow I_{im_{IM_a}}(H_{int_{m_{dep}}})$
10:             $I_{im_{mod_{aff}}} \leftarrow Affect(I_{im_{m_{dep}}})$
11:             if $I_{im_{mod_{aff}}} \neq \varnothing$ then
12:                 $Strategy_{neg} \leftarrow True$
13:             end if
14:         end if
15:     end for
16: end procedure
```

We use the 30 negation markers from (Reitan et al., 2015) for finding negation scope in tweets (shown in Table 4.3). We first detect whether a negation marker appears in either H$_{int}$ or I$_{im}$, but not in both. The reason we also look at I$_{im}$ is to account for negative praise, or ironic blame (Burgers et al., 2012), as in the example "Kayla's cousin **isnt attractive**" → "Kayla's cousin is **attractive**". If the marker is used, we extract its parent node from the dependency tree and if this node is also present in the other utterance then *Negation* strategy is selected [line 4-7 in **Algorithm 2**]. For instance, in "wow **looks** just like me" → "that does **not look** like me", the negation **not** is modifying the main predicate **looks** in H$_{int}$ which is also the main predicate in I$_{im}$ (words are lemmatized). In the next section, we discuss if the parent nodes are not the

aint, cannot, cant, darent, didnt, doesnt, dont, hadnt, hasnt, havent, havnt, isnt, neither, never, no, nobody, none, nor, not, nothing, nowhere, mightnt, mustnt, neednt, oughtnt, shant, shouldnt, wasnt, wouldnt, *n’t

Table 4.3: Negation Lexicons

same but similar and with different sentiment strength (“love” → “**don’t** like”). This strategy is denoted as “Simple_neg” in Table 4.4.

Weakening the intensity of sentiment: the strategy of replacing lexemes expressing a high degree of positive/negative sentiment with more neutral ones is applied only in conjunction with the negation strategy. In the pair “I **love** being sick …” → and “I **don’t like** being sick …”, the negation *don’t* modifies the predicate **like** in H$_{int}$, which is aligned to the word **love** in I$_{im}$, a word with different sentiment strength [shown in line 9-12 in **Algorithm 2**]. We measure the difference in strength using the Dictionary of Affect in Language, respectively introduced in Whissell et al. (1986). Out of 31 I$_{im}$-H$_{int}$ pairs in the dev set, we automatically identify 28 interpretations that are using this approach. In the second strategy, in case of removing the intensifier, we first look whether the intensifier exists in I$_{im}$ and is eliminated from H$_{int}$. Here we use the list of intensifiers from (Taboada et al., 2011). We allow only adjectives and adverbs as intensifiers to discard terms such as “so” since it is a preposition in “…no water **so** I can’t wash …super!!!”. This strategy is used in conjunction with both *lexical antonym* and *Negation* strategies. This algorithm is shown in **Algorithm 3**. For a candidate I$_{im}$-H$_{int}$ pair, if the *lexical antonym* strategy is selected and a_S and a_H are the lexical antonyms, we look whether any intensifier modifies a_S and no intensifier modifies a_H [line 4-5 and line 12-16 in **Algorithm 3**]. In “…I am **really** happy” → “…I am disappointed.”, the intensifier ‘**really** is modifying “happy” (a_S) whereas no intensifier appears in the H$_{int}$ that modifies “disappointed”(a_H).

In case of multiple intensifiers, such as in “**so much** fun …”, the intensifier “so” is modifying the predicate “fun” via another intensifier “much”. Here, we traverse the dependency tree to identify such indirect modifications.

Algorithm 4 $Inten_{anto}$

```
procedure ($I_{im}$, $H_{int}$, $I_{im_{dep}}$, $H_{int_{dep}}$, $anto_{I_{im}}$, $anto_{H_{int}}$)
    $i_{list} \leftarrow intensifiers()$
    $int_{H_{int_{dep}}} \leftarrow int(anto_{H_{int}}, i_{list}, H_{int_{dep}})$
    for $i \leftarrow 0, i_{list}$ do
        $int_{I_{im_{dep}}} \leftarrow int(anto_{I_{im}}, i, I_{im_{dep}})$
        if $int_{I_{im_{dep}}} \neq \varnothing$ & $int_{H_{int_{dep}}} = \varnothing$ then
            $Strategy_{anto_{int}} \leftarrow True$
        end if
    end for
end procedure

procedure $int(node$,$i$,$I_{im_{dep}})$
    if $node \leftarrow I_{im_{dep}}[i]$ then
        Return$True$
    end if
end procedure
```

Algorithm 5 $Inten_{neg}$

```
procedure ($I_{im}$, $H_{int}$, $I_{im_{dep}}$, $H_{int_{dep}}$,$align_{I_{im},H_{int}}$)
    $i_{list} \leftarrow intensifiers()$
    $H_{int_{m_{dep}}} \leftarrow H_{int_{dep}}[neg]$
    $int_{H_{int_{dep}}} \leftarrow int(H_{int_{m_{dep}}}, i_{list}, H_{int_{dep}})$
    $H_{int_{a_{dep}}} \leftarrow align(I_{im_{dep}}[H_{int_{m_{dep}}}])$
    for $i \leftarrow 0, i_{list}$ do
        $int_{I_{im_{dep}}} \leftarrow int(H_{int_{a_{dep}}}, i, I_{im_{dep}})$
        if $int_{I_{im_{dep}}} \neq \varnothing$ & $int_{H_{int_{dep}}} = \varnothing$ then
            $Strategy_{neg_{int}} \leftarrow True$
        end if
    end for
end procedure
```

Algorithm 4 depicts how we detect the absence of an intensifier in the rephrased utterance, in case of a presence of the negation strategy. If the *Negation* strategy is selected, we identify the negated term in the H_{int} and then search its aligned node from the I_{im} using the word-word alignment. Next, we search in the I_{im} if any intensifier intensifying the aligned term [line 8-9 in **Algorithm 4**]. In "...I grow up... **exactly** $like_1$ you" → "...I am not $like_2$ you", the negation **not** is negating $like_2$ in the H_{int} whereas the intensifier **exactly** is intensifying the word ($like_1$); $like_1$ and $like_2$ are the aligned terms. This strategy is denoted as AN_weaksent in Table 4.5 and Table 4.4.

Antonym/Neg + Interrogative to Declarative Transformation: To capture this strategy we need to determine first if the ironic message was expressed as a rhetorical question. We first follow the hypothesis from Oraby et al. (2016b) who found RQs by searching questions in the middle of an utterance since question followed by text cannot be a typical information seeking question. However, such search in dev dataset resulted in low P/R/F1 scores as respectively: 50%, 47.0% and 48.4% showing there are RQs end with questions exist in our dataset (e.g., "Glad people keep their promises these days ?", "Don't you just looooove fighting with people?"). Thus, to identify RQs we propose a supervised classification setup. We first collect two categories of tweets for the classification; tweets that are labeled with #sarcasm that also contain "?", and information seeking tweets containing "?" (altogether 8K tweets, balanced). Next, we train a binary classifier using SVM RBF Kernel with default parameters. The features are Twitter-trained word embeddings (Ghosh et al., 2015), modal verbs such as "could", "should", pronouns, interrogative words such as "why", "what", negations, and position of "?" in a tweet. We evaluate the training model on the dev data and the P/R/F1 improves to is 53.2%, 65.4%, and 58.6%. We further sub-categorize the RQs into $antonym_{rq}$ ("don't you **love** fighting?" → "I **hate** fighting", $negation_{rq}$ ("why am I so photogenic? → I am **not** photogenic'). Once we identify that the ironic message was expressed as a rhetorical question, we identify the specific interpretation strategy accompanying the transformation from interrogative to declarative form: antonym or negation. These combined strategies are denoted as $\mathrm{AN}_{I \rightarrow D}$ in Table 4.5 and Table 4.4.

Desiderative Construction: Automatically identifying desiderative constructions is a complex task. Currently we use a regular expression "I [w]$*$ wish" to capture counterfactuals. This strategy is denoted as AN_desiderative in Table 4.5 and Table 4.4.

When the *Simple negation* and *lexical antonyms* strategies are combined with other strategy (e.g., removing of intensifier), we consider this combined strategy for the interpretation of verbal irony and not the *simple negation* or *lexical antonym* strategy (i.e., we do not double count).

Phrasal antonyms and pragmatic inference: Identifying phrasal antonyms and pragmatic inference is a very hard task, and thus, we propose a method of phrase extraction based on Statistical Machine Translations (SMT). This method is used after all the above strategies are selected. We use an unsupervised alignment technique IBM Model 4 with HMM used in Giza++ (Och and Ney, 2000) to run over the bitexts.[4] For phrase extraction we used Moses (Koehn et al., 2007b) and the IRST language model tool integrated in Moses to build the required language models. Length asymmetry between parallel data (i.e., in textual entailment research hypothesis (H) is usually shorter than the text(T)) is a problem in monolingual alignment research. However, for our data, length asymmetry is not a problem since we instructed the Turkers to consider the entire message while re-phrasing. For I_{im}-H_{int} bitext for example, respectively, the average original ironic message length is 14.56 tokens and the intended meaning message length is 13.26 tokens. Another reason we selected SMT-type alignment methods is the lack of annotated word alignment data (Bar-Haim et al., 2006; Cohn et al., 2008) used generally in monolingual alignment work (Thadani and McKeown, 2011; Yao et al., 2013). In addition, the noisiness of the tweeter messages makes the use of syntactic parsers — employed in monolingual alignment on RTE-type data — less useful. And last but not least, our datasets contain re-orderings where SMT-type alignments are particularly useful. We conducted two alignment experiments; between the (1) H_{int}-H_{int} bitext, and (2) I_{im}-H_{int} bitext. As post-processing, we first remove phrase pairs obtained from the I_{im}-H_{int} bitext that are also present in the set of extracted phrases from the H_{int}-H_{int} bitext. This increases the likelihood of retaining semantically opposite

[4] We used unsupervised alignment mainly due to the lack of annotated word alignment (Thadani and McKeown, 2011).

Strategies	I_{im}-H_{int}	$SIGN$
Lex_ant	2,198 (40.0)	9,691 (51.8)
Simple_neg	1,596 (29.1)	3,827 (20.5)
AN_weaksent	895 (16.3)	2,160 (11.6)
$AN_{I \rightarrow D}$	329 (6.0)	933 (5.0)
AN_desiderative	92 (1.7)	86 (0.5)
AntPhrase+PragInf	357 (6.5)	1912 (10.1)

Table 4.4: Distribution of interpretation strategies on two datasets (numbers and %)

phrases, since phrases extracted from the H_{int}-H_{int} bitext are more likely to be paraphrastic. Second, based on the translation probability scores ϕ, for phrase e if we have a set of aligned phrases f_{set} we reject phrases that have ϕ scores less than $\frac{1}{size(f_{set})}$. The resulting number of phrases extracted from the I_{im}-H_{int} bitext is 11,200. Since we have not manually evaluated these phrase pairs, we only use this set after we have tried all the above strategies. This strategy is denoted as AntPhrase+PragInf in Table 4.5 and Table 4.4.

4.5.2 Results and Distribution of Linguistic Strategies

	dev			$test$			$SIGN_{test}$		
Strategies	P	R	F1	P	R	F1	P	R	F1
Lex_ant	89.0	95.7	92.2	97.2	89.9	93.4	89.4	97.9	93.5
Simple_neg	92.0	89.4	90.7	88.3	88.3	88.3	93.3	91.2	92.2
AN_weaksent	93.6	87.9	90.7	95.0	91.9	93.4	93.3	87.5	90.3
$AN_{I \rightarrow D}$	53.1	65.4	58.6	80.0	0.44	57.2	85.7	70.6	77.4
AN_desiderative	100.0	92.9	96.3	100.0	100.0	100.0	100.0	66.7	80.0
AntPhrase+PragInf	86.2	53.2	65.8	70.7	85.3	77.4	89.5	68.0	77.3

Table 4.5: Evaluation of Computational Methods on dev, $test$ and $SIGN_{test}$ set (in %)

We first report the results of our computational models on the two test sets: $test$ and $SIGN_{test}$ (Table 4.5). Two annotators annotated the $test$ and $SIGN_{test}$ and the agreement between the annotators for both sets is very high (κ > 0.9). We observe that the performance of the models is similar on both $test$ and $SIGN_{test}$ sets, showing consistently good performance (Table 4.5; 90% F1 for all strategies, except the AntPhrase+PragInf and $AN_{I \rightarrow D}$). We conducted a thorough error analysis and have the following observations.

- In some cases the tokenizer was not perfect, especially in words such as “isn’t” the tokenizer missed the negation and that effect the final outcome.

- Tweets often contain words with alternate spellings. The preprocessing strategies handle majority of such cases but still in some cases the tokenizer fails. For instance, the interjection pair "whoo" → "boo" is used with different spellings.

- The sentiment is expressed implicitly in some examples (i.e., exclamative construction). For instance, for the I$_{im}$-H$_{int}$ pair, "what a kick!" → "what a bad kick", the model could not find the Lex_ant strategy since the positive sentiment, e.g., "good" is implicit in the I$_{im}$.[5]

- Antonyms and negations are sometime presented via the hashtag as a combination of multiple words. Such as for "#sweet" → "#notsweet". The model missed these cases. In future, we plan to utilize a split on the hashtags to break it in multiple words, when necessary.

- For the Simple_neg category, we observe in some cases the classifier wrongly identified the presence of a negated term as "Simple_neg", even when negation is present in both I$_{im}$-H$_{int}$ pair. This results from wrong alignment or the classifier could not identify the negated terms in both I$_{im}$ and H$_{int}$. Also, the model mistakes by predicting "never" as a negation where "never" was used as opposite of "always".

- In case the utterance contains multiple sentences, the dependency trees and/or the alignment sometime fail. Thus, in case of multiple strategies the model fails to identify all the strategies.

- The low accuracy for the AN$_{I \to D}$ shows that it is a hard task to find a rhetorical question particularly because not all utterances with question in between depict a rhetorical question. This is also true for the AntPhrase+PragInf category since there is no particular set of rules available that can handle all the different (possible) cases for pragmatic inferences or antonym phrases.

[5] This example is regarding kicking in football.

Given these results, we can now apply these models to study the distribution of these strategies in the entire datasets (Table 4.4). This allows us to do a comparative analysis on a larger scale. We observe the similarities of the distribution of strategies between our dataset I_{im}-H_{int} and $SIGN$ dataset, and also they match very closely to the distribution on the manual annotation on the dev dataset.

Notice that the sum of strategies exceeds the total number of I_{im}-H_{int} pairs for each language since a Tweet can contain several ironic sentences that are interpreted by Turkers. For instance, in "Dave too **nice** . . . a **nice** fella" → "Dave not nice . . . a mean fella" we observe application of two strategies, *lexical antonym* (i.e., **nice** → **mean**) and *negation* (i.e., **nice** → **not nice**).

4.6 Discussion: Hearer-dependent Interpretation Strategies

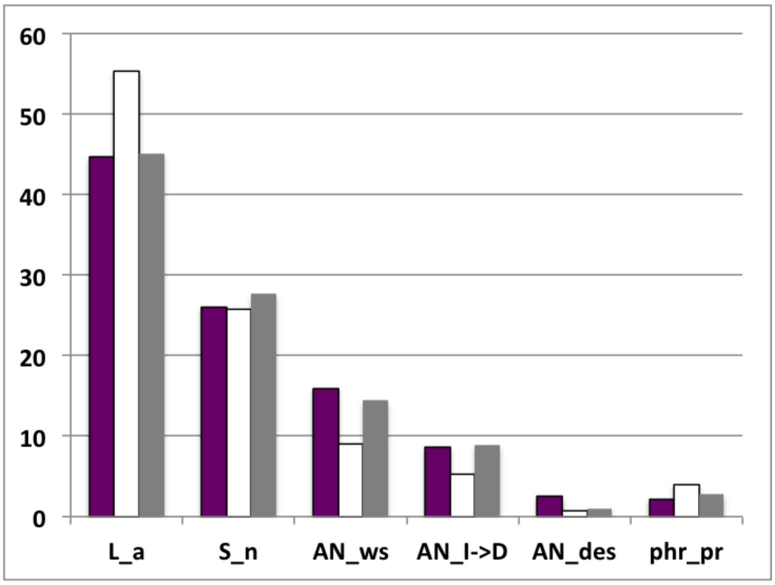


Figure 4.1: Strategies (in%) selected by top three Turkers who attempted 500 HITs (best in color)

In this section we investigate whether hearers adopt similar strategies for interpreting the speaker's ironic nessage. To carry the study we need to analyze interpretation strategies by a set of Turkers who have all re-phrased the same verbal ironic messages I_{im}. From our crowdsource data, we selected three Turkers who have successfully completed the most numbers of HITs (i.e., around five hundred). Since we do not have the annotators' information from the $SIGN$ dataset (Peled and Reichart, 2017), we cannot carry this experiment on it. Figure 4.1 shows the comparison of Turkers' strategies. The turkers are represented as purple (H^1), white (H^2), and grey (H^3) color bars. We refer the three Turkers as H^1, H^2 and H^3, and they are represented by purple, white and grey color respectively in Figure 4.1. For brevity, *L_a* is lexical antonym, *S_n* is simple negation, *AN_WS* is weakening sentiment, $AN_{I\rightarrow D}$ is the Antonym/Neg + Interrogative to Declarative Transformation strategy, *AN_des* is counterfactual, and *phr_pr* is phrase and pragmatic inference strategy. Although the three Turkers choose *lexical antonym*, and *simple negation* as two top choices, there is some variation among the workers. In Figure 4.1, two Turkers (H^1 and H^3) choose *antonyms* more frequently than *negation* while Turker H^2 choose *negation* more than *antonyms*. In Table 4.1, H^i_{int} are generated by the correspondent turker H^i and observe consistently H^2 have chosen the *negation* strategy (even if sometime combined with *weakening of sentiment*). The Figure 4.1 also show that all the three Turkers have chosen the remaining strategies with similar frequencies.

4.6.1 Graded Interpretation:

As touched upon by Giora (1995), antonyms and direct negation are not semantically equivalent strategies, since the second, allows a graded interpretation. Lets look at the examples from Table 4.1 again.

I_{im} : "Pictures of you holding dead animal carcasses are *so flattering*"

H^1 : "Hate hunting season and the pictures of you holding dead animal are *so gross*"

H^2 : "Pictures of you holding dead animal carcasses is an *unflattering look*"

H^3 : "Pictures of you holding dead animal carcasses are *not flattering*"

if "x is not flattering", it is not necessarily bad, but simply "x is less than flattering". Such an implicature is available with sentiment words that allow mediated contraries (Horn, 1989). Direct negation with sentiment words implies that just one value in a set is negated, while the others are potentially affirmed; the scope of the set of possible

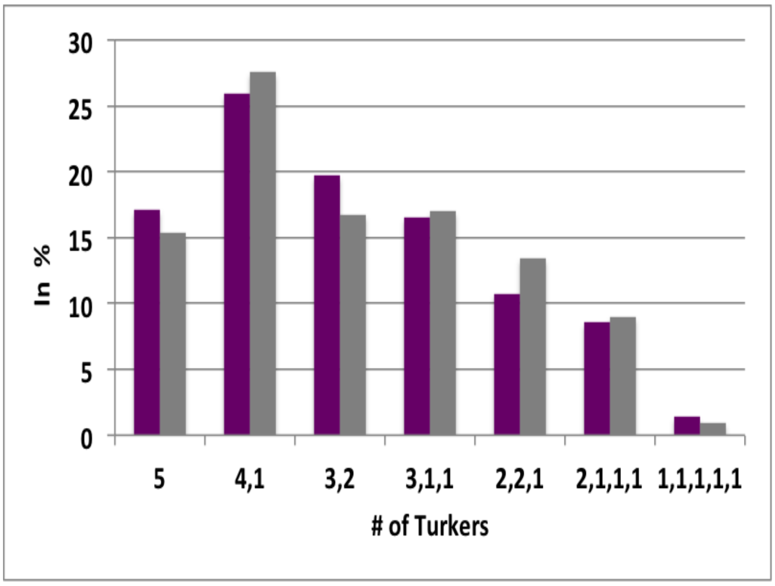


Figure 4.2: Strategies selected per message (in %)

values is larger with words that express a strong sentiment than with words that express a weak one. In other words, the bigger the contrast between the literal and intended meaning, the more ironic the utterance is (Burgers, 2010). In Table 4.1, the I_{im}-H_{int} pair "flattering" → "so gross" and "flattering" → "unflattering look" (interpretation of H^1 and H^2) have more contrast than the pair "flattering" → "not flattering" (interpretation of H^3). As a consequence, H^1 and H^2 perceive the intensity of negative sentiment respectively towards the "picture of dead animals" (target of irony) higher than Turker H^3 (Table 4.1).

4.7 Discussion: Message-dependent Interpretation Strategies

In this section, we investigate whether "the way verbal irony is expressed influences the choice of interpretation strategy by hearers". We looked at I_{im} level distribution of interpretation strategies taken by the hearers for the same given ironic message I_{im}. In Figure 4.2, the vertical columns (e.g., purple represents our dataset I_{im}-H_{int} dataset and grey represents the $SIGN$ dataset) depict the distribution (in %) of tweets strategy-wise. For instance, in our I_{im}-H_{int} dataset, for 17% of messages all five Turkers use the same strategy to interpret the verbal ironic meaning (leftmost on the X-axis labeled as *5*), whereas for 26%, 4 Turkers used same strategy (labeled as *4,1* on X-axis) and so on.

4.7.1 Message Interpreted the Same by All Hearers:

A total of 124 and 433 I_{im}s for I_{im}-H_{int} and $SIGN$ datasets, respectively, have all five interpretations using the same strategy. Looking at the strategies, although Turkers predominantly use *lexical antonyms* (e.g., 68 and 325 for I_{im}-H_{int} and $SIGN$ dataset, respectively) in some cases, all Turkers use RQ or multiple strategies such as both *weakening of sentiment* and *simple negation* together. For the latter (i.e., multiple strategies) majority of the tweets are longer and contain more than one sentence.

Next, we turn to analyze the tweets that express the verbal irony to understand what prompts the Turkers to choose the same strategy and what that informs us (we carry this analysis on our dataset only). We observe *lexical antonyms* strategy is largely used (i.e., $> 90\%$ of times) when the I_{im} was marked by strong subjective words (e.g., “great”, “best”, “prettiest”; we use MPQA lexicon to identify the strength of the sentiment words). These words have been replaced in 90% of cases as lexical antonyms (i.e., “great” → “terrible”). In addition, the majority of adjectives are used in attributive position (i.e., “my **lovely** neighbor is vacuuming at night”), thus, blocking paraphrases involving predicate negation. However, not all strong subjective words guarantee the use of direct opposites in the H_{int}s (e.g., “flattering” → “not flattering”; See Table 4.1). Employing strategies also depend upon the target of ironic intent and how strong the ironic situation is (Ivanko and Pexman, 2003).

Riloff et al. (2013) employed a bootstrapping algorithm to identify “negative situations” from ironic tweets. Their algorithm starts from a seed word (i.e., word that represents positive sentiment) and then based on the proximity of other words the algorithm selects ironic situations, also denoted as the “negative situations” . We use the same algorithm to identify the “negative situations” in the utterances and identify particular situations that are highly correlated to Turker’s choice of *Lexical antonym* strategy. Table 4.6 shows the clusters of the negative situations.

We see in Table 4.6, for instance, utterances containing stereotypical negative situations regarding *health issues* (e.g., “having migraines”, “getting killed by chemicals”) have almost always interpreted with *lexical antonyms* strategies. Also, other undesirable negative states such as “oversleeping”, “waking up early morning”, “luggage lost”,

Cluster Type	Negative Situations
health issues	having migraines, permanent knee damage, getting killed by chemicals, knocking my ankle, waking up with swollen eye, headache, waking up with cut tongue
experiences/state of mind/events	stress in life, oversleeping, homework on Sunday, waking up early starting shitty week, dog food (in cafe), luggage lost, working on Christmas
family/friends	housemate is attention seeker, fake girls, friends ignoring, mom is shouting neighbor is vacuuming late night, no followers
entities	Alabama defense, drivers in Detroit, weather in Scotland

Table 4.6: Stereotypical negative situation clusters

"stress in life" are always interpreted via *lexical antonym* strategy.

Looking at the utterances with *simple negation*, if negative particles are positioned in the ironic message with a sentential scope (e.g., "not a biggie", "not awkward") then they are simply omitted in the interpretations. This trend can be explained according to the inter-subjective account of negation types Verhagen (2005). Sentential negation lead the addressee to open up an alternative mental space where an opposite predication is at stake.

4.7.1.1 Relation between Difficulty of the Interpretation Task and Linguistic Strategies

In the MTurk rephrasing task description we allowed Turkers to self-report how difficult they thought the task was. Based on the number of the strategies selected, we analyze whether there is any correlation between the self-reported difficulty judgments and the strategies used. Table 4.7 represents the % of difficulty judgment given by the Turkers for each interpretation strategy. For instance, when Turkers rephrase the messages with *Lexical Antonyms* strategy, 48.3% of the times they reported the HITs as easiest (i.e., rating= 1) and only 26.1% of the times they reported the HITs as very difficult (i.e., rating = 5). Likewise, for the AN_desiderative strategy, only 1.3% of times the HITs are reported as the easiest ones vs. 2.7% of times they are reported as very difficult (i.e., rating = 5) and so on.

difficulty	Lex_ant	Simple_neg	AN_weaksent	AntPhrase+ PragInf	AN_I → D	AN_desiderative
1	48.3	24.2	15.4	5.6	5.2	1.3
2	38.8	30.3	17.2	5.9	6.1	1.6
3	35.5	30.4	17.3	6.8	7.7	2.1
4	28.9	38.9	12.8	10.3	6.9	1.9
5	26.1	32.4	14.4	16.2	8.1	2.7

Table 4.7: Task difficulty and linguistic strategies

We found the difficulty judgments are strongly correlated to the *Lexical Antonym* strategy (e.g., r = -0.981, p = 0.003; i.e., significant at the 0.005 level), i.e., more Turkers consider the HITs easy when they interpreted using *Lexical Antonyms*. In contrary, for antonym phrases, counterfactuals, and utterances with RQs, Turkers rated the HITs as more difficult (e.g., respectively, r = 0.91, 0.88, 0.94; p = 0.03, 0.04, 0.02; i.e., significant at the 0.05 level).

4.8 Conclusions

We leveraged a crowdsourcing task to obtain a dataset of ironic utterances paired with verbalization of hearers' interpretations of the speaker's intended meaning. We proposed a typology of linguistic strategies for verbal irony interpretation and designed computational models to capture these strategies with good performance. We presented empirical studies aimed to answer three questions: (1) what is the distribution of linguistic strategies used by hearers to interpret ironic messages; (2) do hearers adopt similar strategies to interpret speaker's ironic intent?; and (3) how do expressions of verbal irony influence the choice of interpretation strategy by hearers?

We also observe that the resulting dataset, i.e., the bitext could be useful for research on textual entailment (paraphrases, contradictions), semantic textual similarity (STS), or text generation (given an ironic message automatically generate its interpretation). In Appendix, we introduce some preliminary research on using the rephrasing dataset for STS research on social media genre.

Chapter 5

The Role of Irony in Detecting Agree/Disagree Relations

5.1 Overview

An increasing portion of information and opinion exchange occurs in online interactions such as discussion forums, blogs, and webpage comments.[1] This type of user-generated conversational data provides a wealth of naturally occurring arguments. As online conversation evolves, the participants tend to agree or disagree with other authors or with topic(s) of the discussion. The ability to automatically detect agreement and disagreement in discussions is useful for understanding how conflicts arise and are resolved (Rosenthal and McKeown, 2015). This is a challenging problem due to the dynamic nature of online conversations, and the less formal, and usually very emotional language used in discussing controversial topics (Abbott et al., 2011).

Arguments are predominantly seen to be instruments of persuasion (Tindale and Gough, 1987). There are many ways to persuade people to think or act in a particular way. One such means of persuasion is to use rhetorical devices to get others to adopt their respective points of view. According to Gibbs and Izett (2005), figurative language such as irony is frequently used as an implicit tool in conversation to capture readers' attention (Gibbs and Izett, 2005). Irony could be used to support or attack someone's prior statement. This observation sets up the primary motivation of the research described in this chapter. We want to investigate whether identifying verbal irony in argumentative dialogue improves the accuracy of identify argumentative relations. Using the naming convention of Rosenthal and McKeown (2015), henceforth, we denote the pair of argument relation - agreement and disagreement - as dis(agreement) relations.

[1] Parts of the research related to argument mining has been published at (Ghosh et al., 2014; Wacholder et al., 2014; Ghosh et al., 2016; Musi et al., 2016).

Turn Type	Argument Relation	Irony Relation	Turn Pairs
P_TURN			**userA:** I read it in the text.
C_TURN	Disagree	Verbal Irony	**userB:** Well , as we can see from other recent posts, your reading comprehension is below high school level. Why do folks think lying enhances their POV ?
P_TURN			**userA:** Today, no informed creationist would deny natural selection.
C_TURN	Agree	Verbal Irony	**userB:** Seeing how this was proposed over a century and a half ago by Darwin, what took the creationists so long to catch up ?

Table 5.1: Ironic messages (C_TURNs) and their respective prior turns (P_TURN) from IAC.

Consider the examples from the Internet Argument Corpus in Table 5.1. We present two pairs of turns where the P_TURN depicts the "previous turn" (or, the context) from userA whereas the C_TURN depicts the "current turn" from userB. In the first row, C_TURN disagrees with P_TURN whereas, in the second row, the C_TURN agrees with the P_TURN. Both C_TURNS are annotated as ironic. Although the first and second C_TURN convey disagreement and agreement, respectively, they do not contain *explicit* lexical cues to carry dis(agreement). For instance, it is common to use lexical cues such as "disagree, agree, yes, no", etc. to mark the argument relation. However, instead of such explicit cues the authors use irony to *implicitly* agree/disagree with the other post. We reckon, for a typical dis(agreement) detection task these are challenging examples (i.e., without explicit cues) for the classification task . However, we also observe from the examples that incongruence appear between the C_TURN and P_TURN ("*I read it* ...") and C_TURN ("*your reading comprehension is below high school level*"). Naturally, if we augment features that identify such incongruence it may assist in identifying the dis(agreement). In this chapter, we address the following research question:

- RQ1: Does modeling irony help in dis(agreement) detection?

Our training dataset is based on the Internet Argument Corpus (IAC). We explore various features based on the state-of-the-art for the related task (i.e., detection of (dis)agreement), such as lexical, sentiment, stylistic, and embedding features. We report this set of features as Arg_{feats} features. Next, we analyze the impact of the features that are traditionally used to detect ironic utterances in discussion forums (Section 3.6). We denote this set of features as $Irony_{feats}$.

This chapter is structured as follows. Details of the data collection are reported in Section 5.2. We discuss the related research, particularly on dis(agreement) detection in Section 5.3. Following that, Section 5.4 describes the features and experiments we conduct to address the research question RQ1.

5.2 Data

We have introduced the IAC corpus in great detail in Chapter 2. IAC consists of posts from discussion forums that are based on different contentious topics, such as debates on politics, religion, and gun control debate. Abbott et al. (2011) selected roughly 10K pairs of posts from the IAC corpus for a Mechanical Turk annotation task. These pairs are structurally similar to the turn pairs (P_TURN and C_TURN) as shown in Table 5.1. Abbott et al. (2011) showed annotators seven turn pairs and asked them to judge dis(agreement) and a set of other measures (e.g., irony, respect/insult, nice/nastiness, etc.). Dis(agreement) was a scalar judgment on an 11 point scale [-5,5] where "-5" indicates high disagreement, "0" indicates no dis(agreement), and "5" denotes high agreement.

We followed two following steps to accumulate the training data. First, we utilized the IAC SQL database to collect the P_TURN and C_TURN pairs.[2] Second, we collect only those P_TURN and C_TURN pairs that are also annotated with the verbal irony/sarcasm label. The labels are presented by a score between 0-1, where "1" means the C_TURN is ironic. We use the average scores from the annotators to decide on the final labels of the dis(agreement) label.

Similar to the prior research on this corpus (Rosenthal and McKeown, 2015; Misra and Walker, 2013) we converted the scaler values to three categories: values between [-5, -2] as $disagree$, values between [-1,1] as $none$, and values between [2,5] as $agree$. The number of annotations per category is shown in Table 5.2. Similar to the naming

[2] IAC SQL database is a relatively new version of the IAC corpus. This version is larger than the previous releases.

dis(agreement) types	Irony	# of instances (in %)
$agree$	I	315 (33%)
	NI	638 (67%)
$none$	I	2285 (44.5%)
	NI	2841 (55.6%)
$disagree$	I	2207 (57%)
	NI	1696 (43%)

Table 5.2: Dis(agreement) categories with irony/non-irony data (in %)

convention in the previous chapters, verbal irony is denoted as I and non-irony as NI.

From Table 5.2 we observe (1) the number of P_TURN and C_TURN pairs is low for $agree$ category compared to the other two categories, and (2) the number of I is more for the $disagree$ category. However, overall it is still comparable to the $none$ category.

The large number of ironic instances for the $disagree$ category is not surprising. Sarcasm is typically used to mock others. In argumentative forums, particularly in contentious online forums, such as, IAC, it is unsurprising that users often post sarcastic comments towards others using harsh negative tone. In contrary, the presence of verbal irony for the two remaining categories, $none$ and $agree$ depicts that in many cases, verbal irony can serve a face-saving function, making the speaker appear less rude, particularly when expressing a trivial criticism (Jorgensen, 1996).

5.3 Related Work

The research described in this chapter is related to two areas of studies. First, research in dis(agreement) detection and second, the role of irony in argumentation. We first discuss the research on dis(agreement) detection here.

Prior research on recognition of dis(agreement) relation has focused on spoken dialogue (Galley et al., 2004; Hillard et al., 2003) and only recently, researchers have turned to investigate dis(agreement) in online discussions, especially for online debates (Rosenthal and McKeown, 2015; Abbott et al., 2011; Misra and Walker, 2013; Mukherjee and Liu, 2013; Wang and Cardie, 2016). Although early work have conducted a 2-way (i.e., agreement vs. disagreement) classification, recent studies (Rosenthal and McKeown, 2015; Misra and Walker, 2013) focus into a 3-way classification task (i.e., agreement vs. disagreement vs. none). The majority of the related research utilize lexical (n-gram), sentiment (e.g., dictionaries such as SentiWordNet or MPQA (Wilson et al., 2005)), thread-structures (whether the post is a *root* in the forum), lexicons such as lists of agreement and disagreement terms, to name a few. For instance, apart from the n-gram features, Abbott et al. (2011) investigated dependency relations whereas Misra and Walker (2013) study the topic-independent features (e.g., discourse cues) indicating dis(agreement).

Besides the lexical and sentiment features, Mukherjee and Liu (2013) developed an SVM+ Topic Model classifier to detect (dis)agreement using 2,000 posts. Stab and Gurevych (2014) proposed a suite of lexical, structural (i.e., position of the argument in a paragraph), semantic (i.e., word-embedding based) features for detecting support/attack relations in arguments. Wang and Cardie (2016) used a domain-dependent sentiment lexicon to identify dis(agreement) relations from IAC. In our research, we utilize majority of the state-of-the-arts features in dis(agreement) relations as well as features that are traditionally used in irony detection (e.g., various irony markers, LIWC features, alteration of sentiment between P_TURN and C_TURN, etc).

Based on the structure of threaded discussions in online forums, dis(agreement) recognition can be categorized into three subcategories. First, Yin et al. (2012) focused on the issue of global (dis)agreement that occurs between a post and the root post of the discussion. They annotated 818 posts from the US Message Board and 170 posts from different political forums. Second, the global (dis)agreement, the majority of the research (Abbott et al., 2011; Misra and Walker, 2013; Rosenthal and McKeown, 2015) investigated the quote-response pairs of discussion forums (e.g., Internet Argument Corpus (IAC)) to detect dis(agreement). One important feature in this IAC forum is that is has a mechanism for quoting another post. The author of a response post (i.e., represented as C_TURN in Table 5.1) may decide to quote a previous post (i.e., P_TURN) during reply. IAC corpus contain around 10K such pairs where the majority of the relations are either disagreement or none. Finally, the third category of dis(agreement) relations is described in our prior research (Ghosh et al., 2014). Here, we proposed an annotation scheme for argumentation based on the concept of Callout and Target (introduced in the work of Aakhus et al. (2013)). A *Callout* is a subsequent action that selects (i.e., refers back to) all or some part of a prior action (i.e., Target) and comments on it in some way. A *Target* is a part of a prior action that has been called out by a subsequent action. The corpus consists of blog comments posted as responses to four blog postings selected from a dataset crawled from Technorati between 2008-2010.[3] The experts' annotation task was to identify expressions of Callout and their Targets while also indicating the dis(agreement) between them. Dis(agreement) can occur anytime in a conversation and thus, we consider the Callout-Target framework is a more generic framework that can represent any dis(agreement) relation in a conversation and not only the dis(agreement) between two consecutive posts (i.e., IAC). However, blog comments from the Technorati corpus do not have irony annotations and thus, we do not use the Technorati corpus in this chapter.

In terms of corpora, other than the online forums, Opitz and Zirn (2013) detected the dis(agreement) on sentences from Wikipedia discussion forums. Rosenthal and McKeown (2015) also studied the Wikipedia talk pages to identify dis(agreement). Political debates, such as the Congressional floor-debates (Thomas et al., 2006) is also used to detect the nature of the argument relations. However, the genre of the language of political debates is so different from the informal discussion forums that the results are not directly comparable. The closest to the quote-response structure of IAC corpus is the Usenet forum used in Wang and Rosé (2010). However, as Abbott et al. (2011) discussed, instead of dis(agreement) detection, Wang and Rosé (2010) detected a parent post given a reply post in a forum.

Some related research is involved in stance classification (Walker et al., 2012a; Somasundaran and Wiebe, 2010) that focused on automatic detection of stance of an author with respect to a an issue in debates.

Using irony for dis(agreement) detection is comparatively an unexplored research direction. Although Abbott et al. (2011); Walker et al. (2012a) both have discussed the correlation of verbal irony and disagreement relation the connection is not fully clear.

[3] http://technorati.com/blogs/directory/

For instance, Walker et al. (2012b) mentioned that while one would expect that sarcasm to be positively correlated with disagreement, they did not find such correlation in the IAC dataset. From the theoretical point of view, researchers such as Gibbs and Izett (2005) mention that irony could be used as persuasive strategy especially since irony has the ability to highlight the contrast between expectation and reality. Jorgensen (1996) discusses two different use of irony in online conversation. First, users often complain or mock others in online forums via use of verbal irony, they sometime employ irony as a face-saving function.

5.4 Experiments

We report all the classification experiments and findings in this section. Our goal is to identify dis(agreement) relation in argumentative conversation and to detect whether identification of irony in turn assists in dis(agreement) identification (i.e., addressing RQ1). Our training models are based on response posts (i.e., C_TURN) and the gold labels that indicate whether this post agrees, disagrees, or none, to the post it is replying to (P_TURN). In the next section, we describe the Arg_{feats} features, e.g., features that are used to identify dis(agreement) relations and the classification experiments conducted with the Arg_{feats}. In section 5.4.2 we discuss the role of $Irony_{feats}$ for the same task.

5.4.1 Experiments with Arg_{feats} Features

Lexical Features:

Lexical features are generated for each response posts. They are different n-grams (unigram, bigram, trigram) created based on the full vocabulary of IAC corpus (the cut-off frequency for n-grams used here is five).We also maintain a separate list of n-gram features that capture n-grams from the *first* sentence since the argumentative stance is generally expressed at the beginning of a post. The n-gram features also serve as a strong baseline in Table 5.3.

Argument Lexical Features: In addition, similar to Rosenthal and McKeown (2015), we also used two small lexicons of agreement terms (e.g., “agree”, “accord”; a total of twenty terms) and disagreement terms (e.g., “disagree”, “differ”, “oppose”; a total of nineteen terms) as binary features. This set of features is represented as “arg_lex” in Table 5.3.

Sentiment Features:

Posts that indicate agreement or disagreement to the previous post tend to contain opinions and sentiment (same opinion for agreement and different opinion for disagreement) words. Moreover, posts that are labeled as $none$, tend to be neutral in subjectivity. Thus, subjectivity analysis of the turns will benefit to differentiate between agreement and disagreement. We use (a) the MPQA Lexicon (Wilson et al., 2005) of over 8,000 positive, negative, and neutral sentiment words, and (b) an opinion lexicon with around 6,800 positive and negative sentiment words (Hu and Liu, 2004b) to see whether training instances contain sentiment words. We also include a list of negation terms (around fifty terms) to identify whether any negation is mentioned in the same sentence that contains sentiment term. This set of features is represented as “senti” in Table 5.3.

Hedge Features:

Hedges are linguistic devices used to mitigate the speaker’s commitment to the truth of a proposition (Hyland, 1996), i.e., “I *tend to* accept”. They include possibility modals next to other linguistic items expressing the degree of speaker’s certainty. The use of hedges is common in argumentative posts since they contribute to avoid a potentially face-threatening act of abrupt disagreement. Based on the research of Tan et al. (2016b), we collect a set of candidate hedge cues and use them as Boolean features (presence or absence of a hedge word). Hedges are represented as “hedge” in Table 5.3.[4]

Discourse Features:

Discourse connective features tend to be useful to identify argument relations. For instance, connectives that are collected from the Penn Discourse Tree Bank tend to be used in particular discourse relations to depict *contrast* or *causal* relationship (Prasad et al., 2008). Previous work on dialogue analysis has repeatedly noted the discourse functions of particular discourse markers in argumentative writings (Abbott et al., 2011).

[4] Note, hedge detection is a separate research in NLP (Ganter and Strube, 2009). However, here we are using a curated lexicon that is common in representing hedge.

We use all the discourse connectives from the Penn Discourse Tree Bank as binary features. This set of features is represented as “discours” in Table 5.3.

Modal Verbs:

Modal verbs (e.g., “could”, “should”) work as indicators of argumentative claims since they indicate that what is expressed in a proposition is not unassailable, but might be otherwise (Palau and Moens, 2009; Stab and Gurevych, 2016). We define a Boolean feature which indicates if a C_TURN post contains a modal verb.

Pronoun Features:

Argumentative posts dialogically point to the stance taken by the previous speaker. They, therefore, contain personal pronouns (i.e., “I” see “your” point, etc.). We consider as features both a list of first person (i.e., “I”,“me”, “my”,“mine”) and second person (i.e., “you”, “your”, “youre”) pronouns.

Lexical Entrainment Features:

Rosenthal and McKeown (2015) noted that in argumentative discourse, speakers often tend to mimic the speaking habit of the other person they are talking to. This phenomenon is known as *Entrainment*.[5] We capture accommodation using two features. First, we use Jaccard Similarity to measure lexical similarity between the C_TURN and P_TURN. Next, we also use similarity between the turns using word-embedding vectors (we measure the cosine similarity) of the turns. This set of features is represented as “accod” in Table 5.3.

Embedding Features:

Using the GloVe word embeddings we represent the turns by their average word embeddings (dimension=100) (Pennington et al., 2014). Any word if it does not appear in the GloVe vocabulary, we represent the vector by an *unknown* vector (i.e., a vector based on normal distribution (between 0 to 0.2)). This set of features is represented as “embed” in Table 5.3.

All of the experiments are conducted using SVM (Chang and Lin, 2011). Since, the corpus is heavily unbalanced (refer Table 5.2) we experimented with three particular settings. They are described here.

[5] Note, Rosenthal and McKeown (2015) used the term “accomodation” to denote entrainment.

1. We created a balanced dataset between the three categories by randomly selecting instances from $none$ and $disagree$ to match to the number of utterances in $agree$ category. This resulted in around 950 instances per categories. Since, the number of I and NI for both $none$ and $disagree$ are much more than $agree$, we maintained a 50%-50% balanced between I and NI for $none$ and $disagree$ categories. This is denoted as $arg_{balance}$ in the following section.

2. This time we increased the size of the $none$ and $disagree$ category by selecting n-times of the data of $agree$ category. In other words, $none$ and $disagree$ now have around n*$agree$ instances (still balanced between I and NI whenever possible). This setting is denoted as arg_{imbal_n}.

3. Finally, we used all the data for the three categories ($arg_{imbal_{all}}$).

We present all the results in terms of P/R/F1 scores (per category) as well as the macro-average of the F1 scores for the categories. The datasets are split into 80% training, 10% development, and rest 10% for test. The development data is applied to tune the SVM parameters such as the cost or the regularizer (i.e., L2-regularizer). We experimented with different kernels (e.g., linear, RBF) and finally used the RBF kernel (it performed best with the development set). The development data is also used for a chi-square based feature selection. Notably, we use top-200, top-500, top-1000, top-2000 features in the experiments with the development data and selected top-1000 features since it performed the best. We conduct thorough experiments as well as ablation tests for all the features described above.

Table 5.3 represents the results of $arg_{balance}$. We use two baselines in our experiments - the first is based on n-grams and the second one is based on the average word embedding. Table 5.3 shows the ablation tests for each type of features. We observe that we achieved the best results when using all the features. Likewise, the effect of removal of the sentiment features hurt the F1 scores most. We also observe, removal of the discourse features and hedge words has almost no effect on the final F1 scores. Note, although the overall effect of the removal of hedge words is slight, it affect the $none$ category. This means, speakers' commitment is low in posts categorized with

Features	Categories									
	$none$			$agree$			$disagree$			Average
	P	R	F1	P	R	F1	P	R	F1	F1
n-grams	39.6	22.3	28.6	50.0	**77.7**	60.8	**51.8**	45.7	48.6	45.9
embed	**45.1**	34.0	**38.8**	42.2	37.2	39.5	43.0	58.5	49.5	42.6
Arg_{feats}	38.9	**37.2**	38.0	**62.0**	66.0	**63.9**	48.9	47.9	48.4	**50.1**
Arg_{feats} - arg_lex	41.0	36.2	38.4	54.7	**74.5**	63.1	46.5	35.1	40.0	47.2
Arg_{feats} - senti	38.2	30.9	34.1	55.1	62.8	58.7	46.5	48.9	47.7	46.8
Arg_{feats} - accod	39.7	28.7	33.3	58.0	61.7	59.8	46.5	56.4	51.0	48.0
Arg_{feats} - hedge	40.4	24.5	30.5	**58.6**	69.1	**63.4**	49.1	**59.6**	**53.8**	**49.2**
Arg_{feats} - discourse	41.7	**37.2**	**39.3**	56.6	63.8	60.0	51.1	50.0	50.5	**50.0**

Table 5.3: The effect of various Arg_{feats} for the dis(agreement) categories via ablation tests. P/R/F1 and macro-average (F1) are reported. Best numbers are in **bold**.

$none$.

In order to show the impact of the larger dataset, we increased the size of the $none$ and the $disagree$ category as stated earlier. Figure 5.1 shows the F1 scores of the three categories. Instead of the ablation tests, here, we use all the features since the combination of all features performed the best. Y-axis presents the F1 scores and the X-axis presents the nature of the data used. We observe that the F1 scores of the $none$ category is low initially (around 35%) and then it increases to almost 50% when more data is added. On the other hand, it is interesting to observe that the F1 scores for the $agree$ category remained more or less constant. Initially the F1 score is around 62% and then it remains between 55-60%. We also observe for the $disagree$ category F1 shifts and it reaches the most when we have $disagree$ data that is three-times of the $agree$. However, when we add more data, the F1 score of $disagree$ category drops. Finally, it seems the F1 scores of all the three categories is stabilized (difference between arg_{imbal_4} and $arg_{imbal_{all}}$ is not significant).

5.4.2 Experiments with $Irony_{feats}$ Features

In this section we discuss the effect of the $Irony_{feats}$ in detecting dis(agreement) relations. In Chapter 3, we discussed that several pattern-based, lexical, as well as pragmatic features are used to detect verbal irony in social media (Davidov et al., 2010; Reyes and Rosso, 2011; González-Ibáñez et al., 2011; Riloff et al., 2013; Joshi et al.,

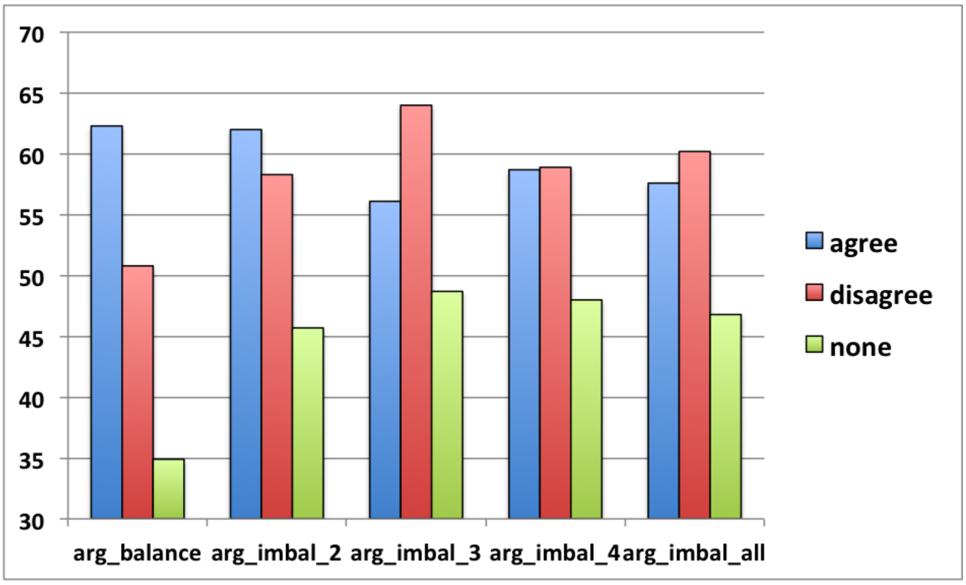


Figure 5.1: F1 scores of dis(agreement) for different settings with Arg_{feats} (better in colour)

2015; Muresan et al., 2016; Wang et al., 2015). We observe that several standard features that are used to detect ironic utterances are also used to identify argument relations. It is unsurprising, since both ironic and argumentative utterances indicate opinions and features that detect subjectivity can help differentiate the type of dis(agreement) or irony. For instance, we observe lexical stylistic features (similar to irony markers; Section 3.5.1) are also exploited in dis(agreement) detection (Rosenthal and McKeown, 2015).

Linguistic Inquirty Word Count Features:

Linguistic Inquiry Word Count (LIWC) features has been used widely in computational approaches to sentiment, emotion, and opinion analysis. These features categorize words into various dictionaries (e.g., linguistic processes such as POS tags, Psychological processes such as emotion, Perceptual processes such as see, hear, personal concerns such as achievement, leisure, etc.). Each C_TURN is represented by the vector of the LIWC dictionaries.

We also computed the cosine similarity between the C_TURN and P_TURN based on their LIWC vector representation. This is similar to measuring the "accommodation" feature (similarity between speaking habits), also used in Rosenthal and McKeown (2015). LIWC features are represented as "LIWC" in Table 5.4.

Sentiment Difference Features:

In conversation, irony is often characterized by the *semantic incongruence* between the C_TURN and P_TURN. We describe this characteristic in detail in Section 3.6. Thus, using the MPQA lexicon, we count the number of positive and negative sentiment tokens, negations, and use a boolean feature that represents whether a C_TURN contains both positive and negative sentiment tokens. We also check whether the current turn C_TURN has a different sentiment than the prior turn P_TURN. (similar to Joshi et al. (2015)).

Irony Markers:

In section 3.5.1 we describe various irony markers we employed to detect ironic utterances from *Reddit* forums and tweets. Here, we utilize similar set of markers, such as tag questions, punctuations, interjections, exclamations, quotation marks, and emoticons as binary features. We utilize a list of tag questions and a list of interjections collected from the grammar sites as binary features. We collected (a) a comprehensive list of emoticons (over one-hundred) from Wikipedia and (b) use standard regular expressions to identify emoticons.[6] Aside from using the emoticons directly as binary features, we use their sentiment as well (e.g., "wink" is regarded as positive sentiment in the MPQA corpus) as features. Emoticons are denoted as "emot" in Table 5.4. Finally, we observe that C_TURN and P_TURN may contain a pair of emoticon that depicts opposite sentiments (i.e., a smily face vs. a sad face). This is also captured as a binary feature. For punctuations (denoted as "punct" in Table 5.4), we keep track of the single use of the punctuation as well as the repeated sequential use (e.g., "!" and "!!!". respectively). Abbott et al. (2011) argued that repeated use of such exclamation mark ("!!!") is different than simple counts of "!" in a C_TURN. We also utilize hyperboles or intensifiers as features because speakers frequently overstate the magnitude of a situation or event. We use terms that are denoted as "strong subjective" (positive/negative) in the MPQA corpus (Wilson et al., 2005) as hyperboles or intensifiers. Hyperboles are represented as "hyper" in Table 5.4.

[6] http://sentiment.christopherpotts.net/code-data/

Features	Categories									
	$none$			$agree$			$disagree$			Average
	P	R	F1	P	R	F1	P	R	F1	F1
n-grams	39.6	22.3	28.6	50.0	**77.7**	60.8	51.8	45.7	48.6	45.9
embed	45.1	34.0	38.8	42.2	37.2	39.5	43.0	**58.5**	49.5	42.6
$A + I_{feats}$	44.6	**47.9**	**46.2**	**68.9**	66.0	**67.4**	51.6	50.0	50.8	**55.0**
$A + I_{feats}$ - LIWC	**44.9**	37.2	40.7	59.8	68.1	63.7	50.5	52.1	**51.3**	52.0
$A + I_{feats}$ - emot	41.6	44.7	43.1	60.2	62.8	61.5	**53.0**	46.8	49.7	51.5
$A + I_{feats}$ - punct	40.2	39.4	39.8	57.4	61.7	59.5	47.2	44.7	45.9	48.5
$A + I_{feats}$ - interj	**45.4**	46.8	46.1	65.9	63.8	64.9	**52.1**	**52.1**	52.1	**54.4**
$A + I_{feats}$ - hyper	43.2	43.6	43.4	**66.7**	68.1	**67.4**	49.5	47.9	48.6	53.2

Table 5.4: The effect of various $Irony_{feats}$ for the dis(agreement) categories via ablation tests. P/R/F1 and macro-average (F1) are reported. Best numbers are in **bold**.

Table 5.4 presents the effect of the $Irony_{feats}$ for dis(agreement) detection via feature ablation tests. For brevity, $A + I_{feats}$ denotes the combined effect of the Arg_{feats} and $Irony_{feats}$. We observe, that $Irony_{feats}$ gives around 6% improvement over the F1 scores when compared to the results from Table 5.3. We also observe, that punctuation features, emoticons, and the LIWC features are more discriminative features for the dis(agreement) detection. Category wise, we found that removal of hyperboles affect the $disagree$ category the most. This is due to the reason that negative hyperbolic terms appear frequently in $disagree$ category. On the other hand, the $agree$ category maintains a F1 in 60% in almost all the ablation tests.

We also look at the $Irony_{feats}$ that are selected via the chi-square selection.

- Interjections such as “please”, “yay”, “huh”, “shoot” are discriminative features for classification.
- Repeated punctuations of exclamation mark (“!”) and question marks (“?”).
- LIWC categories, such as “swear words”, “family”, “pronouns”, “past tense”, etc and WordNet affect category, such as positive emotion.
- Difference in sentiment between C_TURN and P_TURN and emoticons (i.e., smiling emoticon vs. sad-faced emoticon)

In order to show the impact of the larger dataset, Similar to the Figure 5.1 we also experiment with larger dataset with $Irony_{feats}$. This is represented in the Figure 5.2.

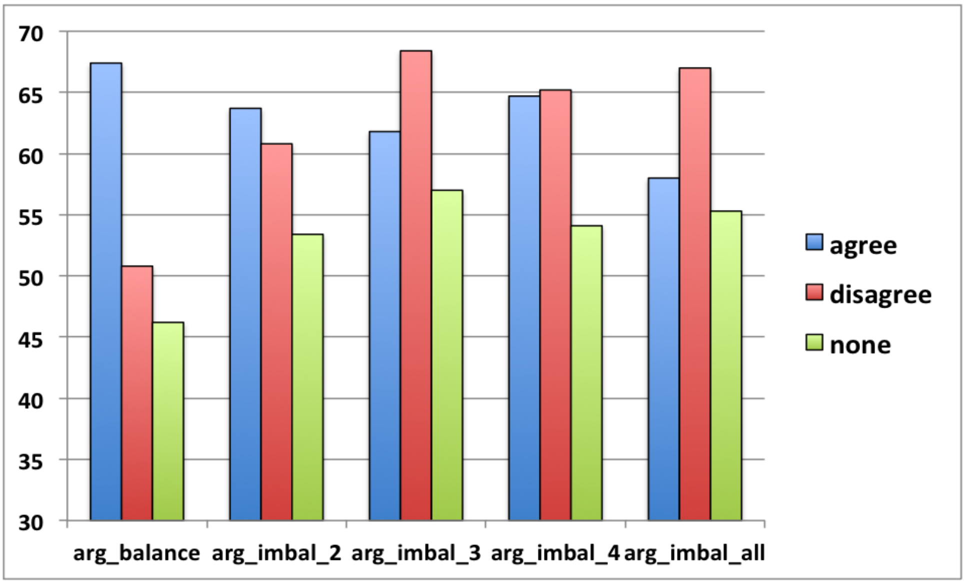


Figure 5.2: F1 scores of dis(agreement) for different settings with $Irony_{feats}$ (better in colour)

Y-axis presents the F1 scores and the X-axis presents the nature of the data used. We observe in general, the performance on the dis(agreement) detection task is better with the $Irony_{feats}$ by around 5% of average F1 score. Similar to the Figure 5.1 the $disagree$ category achieves best F1 with 3*data (of $agree$ category), however, the performance drops (for $disagree$) with addition of more data. At the same time, F1 on $agree$ improves.

5.5 Conclusion

In this chapter we investigate the role of irony features for dis(agreement) detection in online discussion forums. We utilize state-of-the-art features for agree/disagree relation detection on online forums and show with the suitable addition of irony features the performance improves. In future work, we want to investigate the dis(agreement) relationship between irony posts and their subsequent posts. In terms of modeling, we want to utilize the deep learning models we describe in the Chapter 3 (i.e., irony identification) to (a) first detect verbal irony and use the prediction as features, and (b) to employ a multi-learning architecture where we can jointly identify verbal irony as well as dis(agreement) relations.

Chapter 6

Conclusions and Future Work

In this dissertation, we presented empirical studies of verbal irony related to three perspectives of verbal irony, identification, interpretation, and the role of verbal irony in dis(agreement) detection. We performed these studies on different genres of social media: microblogging platform such as Twitter and discussion forums such as *Reddit* and Internet Argument Corpus. During identification of verbal irony in social media we also investigate characteristics of verbal irony, such as *irony markers* (i.e., metacommunicative clues of irony) and *irony factors* (i.e., inherent characteristic of irony that cannot be removed without destroying the irony). We also study utterances, irrespective of any contextual knowledge as well as with context. We particularly showed how conversation context assist in irony identification. Next, we studied how annotators interpret ironic utterances. We conducted thorough user studies and proposed a new taxonomy of linguistic strategies adopted by the annotators while interpreting ironic messages. Finally, in the last chapter we discuss how irony features are useful to detect dis(agreement) in online discussion forums.

In this chapter, Section 6.1 we discuss the main findings of each chapter. Following that, in Section 6.2 we discuss the major limitations of the work presented in this dissertation. Finally, in Section 6.3 we describe the future directions in which the this research can be continued.

6.1 Contributions

Theoretically grounded computational methods: We developed computational models to detect verbal irony that are grounded in theoretical frameworks from Linguistics, Philosophy, and Communication Science. Particularly, we investigate *irony markers* and *irony factors*, two fundamental characteristics of irony (Burgers et al., 2012; Attardo, 2000b; Camp, 2012). We analyze various irony markers (e.g., tropes, morpho_syntactic, and typographic) and showed that for Twitter corpus we achieve an accuracy in 70% while for $Reddit$ the accuracies are in 60%. We also analyze irony markers for different subreddits to show that for contentious forums (e.g., political subreddits) users employ markers more than for forums dedicated on non contentious forums (e.g., technology). Next, we address whether irony markers generalize across different period of time (across different years). We found users consistently utilize similar markers to express verbal irony across different timelines. Chapter 4 is based on developing a new typology of linguistic strategies to categorize annotators' interpretation of ironic messages. Finally, Chapter 5 offers a fresh insight of the use of figurative language such as verbal irony in argument mining research. We showed that in discussion forums often the nature of the dis(agreement) is *implicit* and authors use irony to indicate agreement or disagreement. We showed that adding irony features with dis(agreement) features can improve the accuracy of identification of dis(agreement) by almost 5%.

Novel models for verbal irony identification: We propose new computational models to analyze irony characteristics. First, we propose a reframing of verbal irony detection as a word-sense disambiguating problem: given an utterance, and a target word, identify whether the sense of the target word is literal or ironic. We offer a new Support Vector Machines kernel (based on word embeddings) to identify. the literal vs. ironic sense. This kernel used the *maximum-valued matrix-element* (MVME) algorithm (Islam and Inkpen, 2008) to measure the similarity between utterances via soft-alignment (details in Section 3.5.2.1). We achieve around 7-10% improvement in F1 score over a strong lexical baseline. As another contribution, we utilize a deep learning architecture that separately models the sentiment and ironic situation or context to identify incongruence. For instance, in utterance-level analysis described in Section 3.5.2.2, separate deep learning models are applied on the sentiment and the ironic situation. We use a dual-CNN model where one CNN is applied over the target word (see section 3.5.2.1) and its immediate neighborhood and the other CNN is applied over the rest of the utterance (i.e., the ironic situation). This dual-CNN improves the F1 by 3-4% over the SVM word embedding baseline and around 2% over models that employ a single CNN. Likewise, in conversation analysis, separate deep learning models are applied on conversation turns. Sequence models such as the Long Short-term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are used and we achieve the best accuracy with sentence-level attention model. We also see LSTM performs best when we model the conversation context and ironic reply separately. Here, the incongruence appears between the context (i.e., prior or subsequent posts) and the ironic post and deep learning models that read the posts separately achieve the best results (Section 3.6).

User studies for explaining the prediction of models and for irony interpretations: One question that this dissertation investigates is "what triggers' an ironic reply in a conversation. We conducted thorough user studies in this dissertation to address this question. We compare (a) computational models with (b) human annotations to investigate their agreement in identifying what part of the context can trigger an ironic reply in a conversation. This study is related to the question of explainability of computational models. We looked at the important features (i.e., part of context) predicted by the models and investigated whether humans also recognize the same context that may trigger irony. We compared the attention weights of the LSTM models and observe around 40% of times highest attention weight is given to the sentences (from context) also selected by the annotators (via majority voting). Moreover, attention weights are often correlated to irony characteristic, such as irony factors and irony markers. For instance, given a conversation context and an ironic reply, highest attention is given to the words that present the semantic incongruence between the context and ironic reply. We also examine if an ironic post contains multiple sentences, whether computational models and human annotations identify the same ironic sentences from the post.

In the second user study, we analyze how people interpret irony. We asked the annotators to rephrase ironic utterances to represent the intended meaning. We developed a new typology of linguistic strategies and investigate annotators' behaviors. We empirically validate these strategies over two separate corpora and observe annotators (i.e., Turkers) usually prefer direct lexical antonyms to interpret the intended meaning of the ironic utterances. We also found that annotators' strategies are often based on the shared knowledge between the author of the ironic post and the annotators. For example, for stereotypical ironic situations (e.g., feeling lonely, visiting hospitals, etc.) annotators prefer direct antonyms whereas for irony expressed via rhetorical questions, annotators usually select different strategies to interpret.

Research on diverse datasets: As stated earlier, we studied various corpora for our research. There are two main differences in the datasets. First, tweets are short messages (i.e., maximum of 140 characters) and thus, it is common to use multiple irony indicators to express irony. We achieve very high accuracy on irony identification for Twitter for almost all experiments (i.e., based on irony markers, LISD, etc.). In contrary, discussion posts are long and sometime contain many sentences. We found that often an ironic post may only contain one or two ironic sentences so naturally classification accuracy on IAC or $Reddit$ is lower than Twitter (between 60-70+% for most of the experiments). Another difference in the datasets is based on the availability of labels for the classification task. We observe that data are labeled by two different ways. First, self-annotated, where the users self-labeled their posts (e.g., Twitter, $Reddit$). Second, data is *external-annotated* (crowdsourced) where annotators label the posts (e.g., IAC). In the first case, users use their own perception of irony or sarcasm to label the data whereas in the second case, annotators are provided with the definition of irony and sarcasm. We conducted experiments where we train the machine learning models on one type (e.g., self-labeled) and evaluated on the other (e.g., crowdsourced) and found the performance is lower than when we conduct experiments with the same type of corpus. In the future, we want to conduct experiments on the "same genre" (i.e., debates on similar controversial topics from $Reddit$ and IAC) and evaluate new models. We can also look at the question of domain adaptation.

6.2 Limitations

In this section we discuss the major limitations of the research presented in this dissertation.

This study presents analysis of verbal irony in different social media genre. We looked at short text, i.e., microblogs such as Twitter and also online discussion forums. However, irony is commonplace is every medium of communication and not only social media. Thus, it remains an open question whether the findings from this study will be applicable to other corpora. For instance, in literature, we reckon authors may not utilize *irony markers* since it is more associated with the social media genre.

Second, sarcasm or verbal irony depends to a large extent upon the shared knowledge of speaker/author and hearer/reader (common ground) that is not explicitly part of the ironic utterance or even the conversation context (Haverkate, 1990). When engaged in conversations, speakers might assume that some background knowledge about those topics is understood by the hearers (e.g., historical events, political jargons, etc.). For example, we found ironic posts from *Reddit* that are based on subreddits on video games. Naturally, the authors have shared knowledge on the topic but this is topic specific. A model trained on subreddits on video gamed might not perform well on a subbreddit on political topics. This is also common in Twitter, where authors frequently post ironic or sarcastic tweets assuming the same background knowledge that is shared between the readers of the tweet. For instance, the author of the tweet "shooting in Oakland? that NEVER happens" is expecting that the readers are aware of the city Oakland (it is a city in California) as well as the readers believe that gun violence often occur in Oakland. Identifying ironic utterances with such special background information is a challenging task.

6.3 Future Work

There are many future directions in which to take further the research presented in this dissertation. We summarize some of the major directions below.

In irony identification research we consider the utterances independent of the context and also with the conversation context. Although we looked at immediate context in discussion forums, we can look at the complete thread to study the different topic(s) of discussion to discover how irony evolves. For instance, it is possible that irony is incongruent with the topic of the discussion thread and not incongruent with the particular post of another author.

Second, in the Introduction chapter of the dissertation, we argued that irony needs to be inferred by means of pragmatic interpretation. Computationally this is a challenging problem since often such context is outside the scope of the discussion and we need external knowledge to model them for identifying irony. One possibility is to start with a limited knowledge base of attributes from different entities and mentions. For examples, for cities such as London or Seattle, one of the attributes will be that it rains a lot in these two cites. We can collect such information using Wikipedia or news articles. Next, in ironic utterances if the authors are ironic about those entities ("Sunny days in London") we can compare to the attributes and regard the utterance as ironic.

Third, in terms of collaboration with other areas that study irony we can look at the psycholinguistics and behavioral studies. These studies analyze humans' processing time of literal and ironic utterances. We can utilize a same set of ironic utterance and (a) see the processing time and (b) ask annotators to rephrase (similar to research in Chapter 4). This way we can investigate whether there is any correlation between annotators' rephrasing strategies (e.g., direct antonym, negations, etc.) to the processing time.

Finally, the study of irony as a persuasive strategy can be continued further. We can look at the posts that are direct reply of ironic posts or posts that take place in the forum after an ironic comment. We can look at how irony affect the following discussion in the thread in terms of *topic shift* or *sentiment change*.

Bibliography

Aakhus, M., Muresan, S., and Wacholder, N. (2013). Integrating natural language processing and pragmatic argumentation theories for argumentation support.

Abbott, R., Ecker, B., Anand, P., and Walker, M. A. (2016). Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it. In *LREC*.

Abbott, R., Walker, M., Anand, P., Fox Tree, J. E., Bowmani, R., and King, J. (2011). How can you say such things?!?: Recognizing disagreement in informal political argument. In *Proceedings of the Workshop on Languages in Social Media*, pages 2–11. Association for Computational Linguistics.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics.

Amir, S., Wallace, B. C., Lyu, H., and Silva, P. C. M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Attardo, S. (2000a). Irony as relevant inappropriateness. *Journal of pragmatics*, 32(6):793–826.

Attardo, S. (2000b). Irony markers and functions: Towards a goal-oriented theory of irony and its processing. *Rask*, 12(1):3–20.

Attardo, S., Eisterhold, J., Hay, J., and Poggi, I. (2003). Multimodal markers of irony and sarcasm. *Humor*, 16(2):243–260.

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

Bamman, D. and Smith, N. A. (2015). Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604. Association for Computational Linguistics.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., and Szpektor, I. (2006). The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Barzilay, R. and McKeown, K. R. (2001). Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 50–57. Association for Computational Linguistics.

Becker, H., Naaman, M., and Gravano, L. (2011). Beyond trending topics: Real-world event identification on twitter. *Icwsm*, 11(2011):438–441.

Bharti, S. K., Babu, K. S., and Jena, S. K. (2017). Harnessing online news for sarcasm detection in hindi tweets. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 679–686. Springer.

Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

Bollen, J., Mao, H., and Pepe, A. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *Icwsm*, 11:450–453.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Burgers, C., Van Mulken, M., and Schellens, P. J. (2012). Verbal irony differences in usage across written genres. *Journal of Language and Social Psychology*, 31(3):290–310.

Burgers, C. F. (2010). *Verbal irony: Use and effects in written discourse.* [Sl: sn].

Camp, E. (2012). Sarcasm, pretense, and the semantics/pragmatics distinction*. *Noûs*, 46(4):587–634.

Carvalho, P., Sarmento, L., Silva, M. J., and De Oliveira, E. (2009). Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Chklovski, T. and Pantel, P. (2004). Verbocean: Mining the web for fine-grained semantic verb relations. In *EMNLP*, volume 4, pages 33–40.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Clark, H. H. and Gerrig, R. J. (1984). On the pretense theory of irony.

Cohn, T., Callison-Burch, C., and Lapata, M. (2008). Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614.

Colston, H. L. and O'Brien, J. (2000). Contrast of kind versus contrast of magnitude: The pragmatic accomplishments of irony and hyperbole. *Discourse processes*, 30(2):179–199.

Davidov, D., Tsur, O., and Rappoport, A. (2010). Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10.

De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, pages 449–454. Genoa.

De Marneffe, M.-C., Rafferty, A. N., and Manning, C. D. (2008). Finding contradictions in text. In *ACL*, volume 8, pages 1039–1047.

Dews, S. and Winner, E. (1999). Obligatory processing of literal and nonliteral meanings in verbal irony. *Journal of pragmatics*, 31(12):1579–1599.

Dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Fancellu, F., Lopez, A., and Webber, B. (2016). Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 495–504.

Frank, J. (1990). You call that a rhetorical question?: Forms and functions of rhetorical questions in conversation. *Journal of Pragmatics*, 14(5):723–738.

Galley, M., McKeown, K., Hirschberg, J., and Shriberg, E. (2004). Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 669. Association for Computational Linguistics.

Ganter, V. and Strube, M. (2009). Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 173–176. Association for Computational Linguistics.

Gerrig, R. J. and Goldvarg, Y. (2000). Additive effects in the perception of sarcasm: Situational disparity and echoic mention. *Metaphor and Symbol*, 15(4):197–208.

Ghosh, A. and Veale, T. (2016). Fracking sarcasm using neural network. In *WASSA@ NAACL-HLT*, pages 161–169.

Ghosh, D., Fabbri, A. R., and Muresan, S. (2017). The role of conversation context for sarcasm detection in online interactions. *arXiv preprint arXiv:1707.06226*.

Ghosh, D., Guo, W., and Muresan, S. (2015). 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. Association for Computational Linguistics.

Ghosh, D., Khanam, A., Han, Y., and Muresan, S. (2016). Coarse-grained argumentation features for scoring persuasive essays. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 549.

Ghosh, D. and Muresan, S. (2018). “with 1 follower i must be awesome :p”. exploring the role of irony markers in irony recognition. *Proceedings of ICWSM*.

Ghosh, D., Muresan, S., Wacholder, N., Aakhus, M., and Mitsui, M. (2014). Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 39–48.

Gibbs, R. W. (1986). On the psycholinguistics of sarcasm. *Journal of Experimental Psychology: General*, 115(1):3.

Gibbs, R. W. and Colston, H. L. (2007). *Irony in language and thought: A cognitive science reader*. Psychology Press.

Gibbs, R. W. and Izett, C. (2005). Irony as persuasive communication. *Figurative language comprehension: Social and cultural influences*, pages 131–151.

Gibbs Jr, R. W. (1993). Process and products in making sense of tropes.

Gibbs Jr, R. W. (2003). Nonliteral speech acts in text and discourse. *Handbook of discourse processes*, pages 357–393.

Gibbs Jr, R. W., O'Brien, J. E., and Doolittle, S. (1995). Inferring meanings that are not intended: Speakers intentions and irony comprehension. *Discourse Processes*, 20(2):187–203.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the ACL*, pages 42–47.

Giora, R. (1995). On irony and negation. *Discourse processes*, 19(2):239–264.

Giora, R. (1997). Understanding figurative and literal language: The graded salience hypothesis. *Cognitive Linguistics (includes Cognitive Linguistic Bibliography)*, 8(3):183–206.

Giora, R. and Fein, O. (1999). Irony: Context and salience. *Metaphor and Symbol*, 14(4):241–257.

Giora, R., Fein, O., and Schwartz, T. (1998). Irony: grade salience and indirect negation. *Metaphor and Symbol*, 13(2):83–101.

González-Ibáñez, R., Muresan, S., and Wacholder, N. (2011). Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*, pages 581–586. Association for Computational Linguistics.

Grice, H. P. (1978). Further notes on logic and conversation. *1978*, 1:13–128.

Grice, H. P., Cole, P., and Morgan, J. L. (1975). Syntax and semantics. *Logic and conversation*, 3:41–58.

Guo, W. and Diab, M. (2012a). Learning the latent semantics of a concept from its definition. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 140–144. Association for Computational Linguistics.

Guo, W. and Diab, M. (2012b). Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872. Association for Computational Linguistics.

Gutiérrez, E. D., Shutova, E., Marghetis, T., and Bergen, B. (2016). Literal and metaphorical senses in compositional distributional semantic models. In *ACL (1)*.

Hallmann, K., Kunneman, F., Liebrecht, C., van den Bosch, A., and van Mulken, M. (2016). Sarcastic soulmates: Intimacy and irony markers in social media messaging. *LiLT (Linguistic Issues in Language Technology)*, 14.

Harris, Z. S. (1954). Distributional structure. *Word*, 10:146–162.

Haverkate, H. (1990). A speech act analysis of irony. *Journal of Pragmatics*, 14(1):77–109.

Hidey, C. and McKeown, K. (2018). Persuasive influence detection: The role of argument sequencing.

Hillard, D., Ostendorf, M., and Shriberg, E. (2003). Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2*, pages 34–36. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hong, L. and Davison, B. D. (2010). Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88. ACM.

Horn, L. (1989). *A natural history of negation.* The University of Chicago Press.

Hu, M. and Liu, B. (2004a). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Hu, M. and Liu, B. (2004b). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Hyland, K. (1996). Talking to the academy: Forms of hedging in science research articles. *Written communication*, 13(2):251–281.

Islam, A. and Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(2):10.

Ivanko, S. L. and Pexman, P. M. (2003). Context incongruity and irony processing. *Discourse Processes*, 35(3):241–279.

Jorgensen, J. (1996). The functions of sarcastic irony in speech. *Journal of pragmatics*, 26(5):613–634.

Jorgensen, J., Miller, G. A., and Sperber, D. (1984). Test of the mention theory of irony. *Journal of Experimental Psychology: General*, 113(1):112.

Joshi, A., Sharma, V., and Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China. Association for Computational Linguistics.

Joshi, A., Tripathi, V., Bhattacharyya, P., Carman, M., Singh, M., Saraswati, J., and Shukla, R. (2016). How challenging is sarcasm versus irony classification?: A study with a dataset from english literature. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 123–127.

Justo, R., Corcoran, T., Lukin, S. M., Walker, M., and Torres, M. I. (2014). Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web. *Knowledge-Based Systems*, 69:124–133.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Katz, A. N. and Pexman, P. M. (1997). Interpreting figurative statements: Speaker occupation can change metaphor to irony. *Metaphor and Symbol*, 12(1):19–41.

Kaufer, D. S. (1981). Understanding ironic communication. *Journal of Pragmatics*, 5(6):495–510.

Khattri, A., Joshi, A., Bhattacharyya, P., and Carman, M. (2015). Your sentiment precedes you: Using an authors historical tweets to predict sarcasm. In *Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 25–30.

Khodak, M., Saunshi, N., and Vodrahalli, K. (2017). A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007a). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007b). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL* *on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Kreuz, R. J. and Glucksberg, S. (1989). How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of Experimental Psychology: General*, 118(4):374.

Kreuz, R. J. and Roberts, R. M. (1993). On satire and parody: The importance of being ironic. *Metaphor and Symbol*, 8(2):97–109.

Krippendorff, K. (2012). *Content analysis: An introduction to its methodology*. Sage.

Lamb, A., Paul, M. J., and Dredze, M. (2013). Separating fact from fear: Tracking flu infections on twitter. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 789–795.

LeCun, Y. and Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Lei, T., Barzilay, R., and Jaakkola, T. (2015). Molding cnns for text: non-linear, non-consecutive convolutions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1565–1575, Lisbon, Portugal. Association for Computational Linguistics.

Liebrecht, C., Kunneman, F., and van den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets# not.

Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., and Lei, K. (2014). Sarcasm detection in social media based on imbalanced classification. In *International Conference on Web-Age Information Management*, pages 459–471. Springer.

Maynard, D. and Greenwood, M. A. (2014). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of LREC*.

Méndez-Naya, B. (2008). Special issue on english intensifiers. *English Language and Linguistics*, 12(02):213–219.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Mishra, A., Kanojia, D., Nagar, S., Dey, K., and Bhattacharyya, P. (2016). Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Misra, A. and Walker, M. A. (2013). Topic independent identification of agreement and disagreement in social media dialogue. In *Proceedings of the SIGDIAL 2013 Conference*, pages 41–50. Association for Computational Linguistics.

Mohammad, S. M., Dorr, B. J., Hirst, G., and Turney, P. D. (2013). Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.

Muecke, D. C. (1969). *The compass of irony*. Routledge.

Mukherjee, A. and Liu, B. (2013). Discovering user interactions in ideological discussions. In *Proceedings of the 51st Annual Meeting on Association for Computational Linguistics*, pages 671–681. Citeseer.

Muresan, S., Gonzalez-Ibanez, R., Ghosh, D., and Wacholder, N. (2016). Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology*.

Musi, E. (2017). How did you change my view? a corpus-based study of concessions argumentative role. *Discourse Studies*, page 1461445617734955.

Musi, E., Ghosh, D., and Muresan, S. (2016). Towards feasible guidelines for the annotation of argument schemes. *ACL 2016*, page 82.

Och, F. J. and Ney, H. (2000). Giza++: Training of statistical translation models.

Och, F. J. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449.

Opitz, B. and Zirn, C. (2013). Bootstrapping an unsupervised approach for classifying agreement and disagreement. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 253–265.

Oraby, S., Harrison, V., Hernandez, E., Reed, L., Riloff, E., and Walker, M. (2016a). Creating and characterizing a diverse corpus of sarcasm in dialogue.

Oraby, S., Harrison, V., Misra, A., Riloff, E., and Walker, M. (2017). Are you serious?: Rhetorical questions and sarcasm in social media dialog. *arXiv preprint arXiv:1709.05305*.

Oraby, S., Harrison, V., Reed, L., Hernandez, E., Riloff, E., and Walker, M. (2016b). Creating and characterizing a diverse corpus of sarcasm in dialogue. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 31.

Palau, R. M. and Moens, M.-F. (2009). Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107. ACM.

Palinkas, I. (2013). Irony and the standard pragmatic model. *International Journal of English Linguistics*, 3(5):14.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Partee, B. (1984). Compositionality. varieties of formal semantics: Proceedings of the fourth amsterdam colloquium, ed. by frank veltman, 281–311.

Peled, L. and Reichart, R. (2017). Sarcasm sign: Interpreting sarcasm with sentiment based monolingual machine translation. *arXiv preprint arXiv:1704.06836*.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001a). Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, page 71.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001b). Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

Prabhakaran, V. and Boguraev, B. (2015). Learning structures of negations from flat annotations. *Lexical and Computational Semantics (* SEM 2015)*, page 71.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The penn discourse treebank 2.0. In *LREC*. Citeseer.

Ptáček, T., Habernal, I., and Hong, J. (2014). Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223.

Rajadesingan, A., Zafarani, R., and Liu, H. (2015). Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 97–106. ACM.

Ramage, D., Dumais, S. T., and Liebling, D. J. (2010). Characterizing microblogs with topic models. *ICWSM*, 10(1):16.

Recanati, F. (2004). *Literal meaning*. Cambridge University Press.

Regel, S. (2009). *The comprehension of figurative language: electrophysiological evidence on the processing of irony*. PhD thesis, Max Planck Institute for Human Cognitive and Brain Sciences Leipzig.

Rei, M., Bulat, L., Kiela, D., and Shutova, E. (2017). Grasping the finer point: A supervised similarity network for metaphor detection. *arXiv preprint arXiv:1709.00575*.

Reitan, J., Faret, J., Gambäck, B., and Bungum, L. (2015). Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108.

Reyes, A. and Rosso, P. (2011). Mining subjective knowledge from customer reviews: A specific case of irony detection. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 118–124. Association for Computational Linguistics.

Reyes, A., Rosso, P., and Veale, T. (2013). A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47(1):239–268.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 704–714. Association for Computational Linguistics.

Ritter, A., Cherry, C., and Dolan, B. (2010). Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180. Association for Computational Linguistics.

Ritter, A., Etzioni, O., Clark, S., et al. (2012). Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM.

Roberts, R. M. and Kreuz, R. J. (1994). Why do people use figurative language? *Psychological science*, 5(3):159–163.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., and Blunsom, P. (2015). Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Rosenthal, S. and McKeown, K. (2015). I couldnt agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 168.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM.

Scharrer, L. and Christmann, U. (2011). Voice modulations in german ironic speech. *Language and speech*, 54(4):435–465.

Schifanella, R., de Juan, P., Tetreault, J., and Cao, L. (2016). Detecting sarcasm in multimodal social platforms. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1136–1145. ACM.

Schwoebel, J., Dews, S., Winner, E., and Srinivas, K. (2000). Obligatory processing of the literal meaning of ironic utterances: Further evidence. *Metaphor and Symbol*, 15(1-2):47–61.

Shutova, E. V. (2011). Computational approaches to figurative language. Technical report, University of Cambridge, Computer Laboratory.

Socher, R., Manning, C. D., and Ng, A. Y. (2010). Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9.

Somasundaran, S. and Wiebe, J. (2010). Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124. Association for Computational Linguistics.

Stab, C. and Gurevych, I. (2014). Annotating argument components and relations in persuasive essays. In *COLING*, pages 1501–1510.

Stab, C. and Gurevych, I. (2016). Parsing argumentation structure in persuasive essays. *arXiv preprint, arxiv.org/abs/1604.07370*.

Strapparava, C., Valitutti, A., et al. (2004). Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., and Lee, L. (2016a). Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th international conference on world wide web*, pages 613–624. International World Wide Web Conferences Steering Committee.

Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., and Lee, L. (2016b). Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of WWW*.

Tepperman, J., Traum, D. R., and Narayanan, S. (2006). " yeah right": sarcasm recognition for spoken dialogue systems. In *INTERSPEECH*.

Thadani, K. and McKeown, K. (2011). Optimal and syntactically-informed decoding for monolingual phrase-based alignment. In *ACL (Short Papers)*, pages 254–259.

Thomas, M., Pang, B., and Lee, L. (2006). Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 327–335. Association for Computational Linguistics.

Tindale, C. W. and Gough, J. (1987). The use of irony in argumentation. *Philosophy & rhetoric*, pages 1–17.

Tsur, O., Davidov, D., and Rappoport, A. (2010). Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*.

Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E., and Dyer, C. (2014). Metaphor detection with cross-lingual model transfer.

Tufekci, Z. (2014). Big questions for social media big data: Representativeness, validity and other methodological pitfalls.

Utsumi, A. (2000). Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics*, 32(12):1777–1806.

Veale, T. and Hao, Y. (2010). Detecting ironic intent in creative comparisons. In *ECAI*, volume 215, pages 765–770.

Verhagen, A. (2005). *Constructions of intersubjectivity: Discourse, syntax, and cognition*. Oxford University Press on Demand.

Wacholder, N., Muresan, S., Ghosh, D., and Aakhus, M. (2014). Annotating multiparty discourse: Challenges for agreement metrics. *LAW VIII*, page 120.

Walker, M. A., Anand, P., Abbott, R., and Grant, R. (2012a). Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596. Association for Computational Linguistics.

Walker, M. A., Tree, J. E. F., Anand, P., Abbott, R., and King, J. (2012b). A corpus for research on deliberation and debate. In *LREC*, pages 812–817.

Wallace, B. C. (2015). Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4):467–483.

Wallace, B. C., Do Kook Choe, L. K., Kertz, L., and Charniak, E. (2014). Humans require context to infer ironic intent (so computers probably do, too). In *ACL (2)*, pages 512–516.

Wang, L. and Cardie, C. (2016). Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. *arXiv preprint arXiv:1606.05706*.

Wang, Y.-C. and Rosé, C. P. (2010). Making conversational structure explicit: identification of initiation-response pairs within online discussions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 673–676. Association for Computational Linguistics.

Wang, Z., Wu, Z., Wang, R., and Ren, Y. (2015). Twitter sarcasm detection exploiting a context-based model. In *International Conference on Web Information Systems Engineering*, pages 77–91. Springer.

Weng, J. and Lee, B.-S. (2011). Event detection in twitter. *ICWSM*, 11:401–408.

Whissell, C., Fournier, M., Pelland, R., Weir, D., and Makarec, K. (1986). A dictionary of affect in language: Iv. reliability, validity, and applications. *Perceptual and Motor Skills*, 62(3):875–888.

Williams, R. (1983). Sociological tropes: A tribute to erving goffman. *Theory, Culture & Society*, 2(1):99–102.

Wilson, D. (2006). The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116(10):1722–1743.

Wilson, D. and Sperber, D. (1992). On verbal irony. *Lingua*, 87(1):53–76.

Wilson, D. and Sperber, D. (2002). Relevance theory. *Handbook of pragmatics*.

Wilson, D. and Sperber, D. (2012). Explaining irony. *Meaning and relevance*, pages 123–145.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). 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.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489.

Yao, X., Clark, P., Van Durme, B., and Callison-Burch, C. (2013). A lightweight and high performance monolingual word aligner. In *Proceedings of the 2013 Conference of the Association for Computational Linguistics (ACL 2013)*, Sofia, Bulgaria. Association for Computational Linguistics.

Yin, J., Thomas, P., Narang, N., and Paris, C. (2012). Unifying local and global agreement and disagreement classification in online debates. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 61–69. Association for Computational Linguistics.

Yin, W., Schütze, H., Xiang, B., and Zhou, B. (2015). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., et al. (2014). Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Zhang, M., Zhang, Y., and Fu, G. (2016). Tweet sarcasm detection using deep neural network.

Zou, W. Y., Socher, R., Cer, D., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.