May 2018

The role of approximate negators in modeling the automatic detection of negation in tweets

Norma E. Palomino
Syracuse University

Abstract

Although improvements have been made in the performance of sentiment analysis tools, the automatic detection of negated text (which affects negative sentiment prediction) still presents challenges. More research is needed on new forms of negation beyond prototypical negation cues such as “not” or “never.” The present research reports findings on the role of a set of words called “approximate negators,” namely “barely,” “hardly,” “rarely,” “scarcely,” and “seldom,” which, in specific occasions (such as attached to a word from the non-affirmative adverb “any” family), can operationalize negation styles not yet explored. Using a corpus of 6,500 tweets, human annotation allowed for the identification of 17 recurrent usages of these words as negatives (such as “very seldom”) which, along with findings from the literature, helped engineer specific features that guided a machine learning classifier in predicting negated tweets. The machine learning experiments also modeled negation scope (i.e. in which specific words are negated in the text) by employing lexical and dependency graph information. Promising results included F1 values for negation detection ranging from 0.71 to 0.89 and scope detection from 0.79 to 0.88. Future work will be directed to the application of these findings in automatic sentiment classification, further exploration of patterns in data (such as part-of-speech recurrences for these new types of negation), and the investigation of sarcasm, formal language, and exaggeration as themes that emerged from observations during corpus annotation.

THE ROLE OF APPROXIMATE NEGATORS IN MODELING THE AUTOMATIC DETECTION OF NEGATION IN TWEETS

by

Norma Palomino

Bachelor of Arts in Philosophy, Universidad de Moron (Buenos Aires), 1998
Master of Sciences in Information Studies, University of Texas at Austin, 2003

Dissertation
Submitted in partial fulfillment of the requirements for the degree of
Doctor of Professional Studies in Information Management.

Syracuse University
May 2018

Acknowledgments

This doctoral dissertation is dedicated to the loving memory of my mother

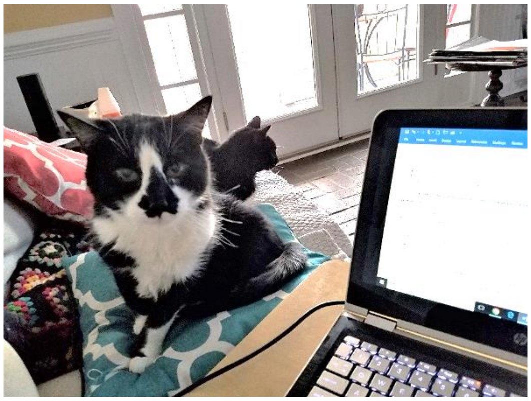
Ofelia Mabel Pisano, for whom I sign it as:

Norma Estela Pisano Palomino

Because the caring power of women should never be hidden in their children's name

It takes a village to raise a child and it takes pretty much the same number of people for a doctoral student to get their research done. Six years of hard work could not have been possible without the patience, support, and love of many, many unforgettable people. Thanks to Dr. Alistair Kennedy for his clarifications and help. Thanks to Briana Galea, the most dedicated annotator in the world, for her hard work, patience, feedback, and explanations on the intricated ways the English language expresses negation. Thanks also to Ever de los Rios for his patience and friendship and magic Excel skills. Special thanks to Nikhil Kini for helping me fix Tweebo for this research, and also for the wonderful guitar videos and sense of humor ("…you should be able to figure it out, easily or eventually," and indeed I did!). Forever grateful to my English language editor Melissa Kizina Motsch, the sweetest angel on earth, editing draft after draft while changing diapers of her newborn (cutest) baby (ever born). I am also forever grateful to my fellow DPS students Dawn Bovasso, Jake Dolezal, Tyson Brooks, Sarah Chauncey, and Barbara Stripping for helping me make up my mind and apply to the program. It was one of the best decisions of my life, and I owe it to you, (also special thanks to Dawn and Jake for storytelling and drinks, and to Sarah for being the best roommate ever). Perhaps one of the most important "thank you" notes goes to Michelle Kaarst-Brown, former Executive Director of the Program, for seeing my potential as a researcher and for supporting me through the process of learning to be a doctor; thanks also to Michelle for your friendship.

Most importantly, I am very, very grateful to my wonderful advisor Nancy McCracken for her research guidance, advice, support, good sense of humor, and cupcakes with frosting-made Argentinean flag toppings: Nancy, you are an outstanding advisor with a warm soul, and I am absolutely honored to have you along in my doctoral journey. Infinite thanks also to my former advisor, Jennifer Stromer-Galley, for helping me in the first stages of my research work: thanks to you Jenny I got the foundational skills for a future scholarship. Special thanks also to Marilyn Arnone for her support and guidance as academic advisor when I joined the DPS program. Finally, very special thanks to Jen Barclay and Sue Nemier for their invaluable support with forms and other doctoral student puzzle-style paperwork.



I'm immensely grateful to life for my furry research and everyday companions Junior (front) and Druh (back)

A warmhearted "thank you" goes to my study buddy and special friend, John Stinnett: John, without your encouragement and support in our weekly touch base meetings, text messages, phone calls, etc., I would had never reached this point.

Endless appreciation to my long-time friends from Argentina and Canada: Olga and Roberto Santander, Natalia Bardoni, Bettina Dibon, Karina Pirillo, Franco Macchiarulo, my sister Laura Palomino, and my cousin Patricia Alvarez Taubin: thanks for your patience in reducing very important conversations to a 20-minute phone call “because Norma has to study...”

Finally, and most importantly: special thanks to the love of my life and best friend Ronald Stuart Grubb Jr. for the many hours of patient and silent support, loving words of encouragement, hundreds of tweets revisions, extra language editing, for witnessing my defense as another random attendant, and for all the infinite love you make me feel every day of my life. This dissertation is a consequence of your loving support and for that, my infinite gratitude to you.

Table of Contents

Chapter 1: Introduction

Chapter 2: Literature Review of Negation in Linguistic and Computational Linguistic Studies

Chapter 3: Research Design

Chapter 4: Data Analysis and Results

Chapter 5: Limitations and Future Work

Figures and Tables

Figures

Tables

Chapter 1: Introduction

Importance of the Study: Current Challenges in Negative Sentiment Detection

Although recent advancements have been made in sentiment analysis in the field of computational linguistics, researchers still face substantial challenges. One such challenge is in how to effectively model negation prediction which, at times, directly affects negative sentiment detection (Blanco & Moldovan, 2011, March; Councill, McDonald & Velikovich, 2010, July). For example, in the 2017 International Workshop on Semantic Evaluation (SemEval), task 4 on Sentiment Analysis in Twitter, two out of the top three solutions for the 5-point sentiment quantification competence (subtask E) have achieved their best performances using focused negation features with the classification algorithm of logistic regression (Balikas, 2017; Li, Nourbakhsh, Liu, Fang, & Shah, 2017). This subtask required automatic ordinal classification of tweets among 5 types of polarities: "VeryPositive," "Positive," "Neutral," "Negative," and "VeryNegative." However, in spite of his solution's competitive performance, author Balikas reported that his model still returned the highest error rate (macro-averaged mean-absolute-error) in the "VeryNegative class," which means that this class was more prone to error than the rest of the classes. Negation modeling, thus, requires further research.

Furthermore, following new trends in product review practices by platforms such as Amazon, TripAdvisor and Yelp, recent editions of the SemEval workshop have concentrated on granularity in sentiment valence (i.e. sentiment intensity) as well as broader forms of affect beyond sentiment, such as distinctive types of emotions[1]

[1] SemEval-2018 Task 1, Affect in Tweets (AIT-2018): https://competitions.codalab.org/competitions/17751

(Rosenthal, Farra & Nakov, 2017). This shift towards differentiating distinctive forms of sentiment and emotion also pushes the research agenda for negative sentiment detection, increasing the need to move away from a binary (i.e. negative and non-negative) approach to negation modeling and towards identifying more operationalizations of negation.

Two focused efforts in modeling negation detection deserve special attention. The first is the shared task dedicated to resolving the scope and focus of negation offered as part of the first edition of the 2012 Joint Conference on Lexical and Computational Semantics (so-called SEM 2012; Morante & Blanco, 2012). This shared task introduced new non-subject specific corpora comprised of Wall Street Journal articles and stories by Arthur Conan Doyle. The corpora were annotated with three negation-related elements: negation cue, cue's scope and negated event. From the twelve solutions presented for the scope resolution track, the winning solution (in the category of strict cue match) introduced semantic aspects of negation and dependency parsing for scope resolution (Lapponi, Read, & Øvrelid, 2012, December).

The second effort was formulated when modeling negation for sentiment analysis in Twitter data. The winning team ("NRC-Canada") in task 2 in the 2013 edition of the abovementioned SemEval workshop employed two SVM classifiers to detect sentiment at the sentence and term levels (Mohammad, Kiritchenko & Zhu, 2013, June). Although their solution trumped those proposed by 44 other teams, their strategy still needed some fine-tuning, especially in the area of negation detection. In a second iteration of their solution (for task 9 of SemEval 2014), these researchers reported that improving negation modeling raised the F-score in all datasets; for example, in a 2013 dataset from Twitter, they reached a metric of 85.19, while measurements taken from the 2014 dataset saw an increase of 86.63. The main improvement that the team made was to apply a more discriminating approach when dealing semantically with negation words (Zhu, Kiritchenko, & Mohammad, 2014, August). As shown by the findings in these reports, better negation modeling definitely improves overall sentiment precision scores, and focusing on semantic features contributes to the model’s effectiveness.

Negated contexts and the standard reversing assumption. It is worth mentioning that the traditional approach in negation modeling, first proposed by Pang, Lee, and Vaithyanathan (2002, July), consists of identifying a negation cue first (i.e. a prototypical negation word such as “no“ or “never“) and then labeling all consecutive tokens after that as negated (attaching a tag such as “NEG“) until the first punctuation mark is found; those tokens are thus considered under the scope of said negation cue. This approach, commonly refereed as “negated contexts,“ is at times based upon an understanding of the performance of negation cues that Zhu, Guo, Mohammad, and Kiritchenko call “standard reversing assumption“ (2014), but it is also called “polarity flip“ or “switch negation“ (Taboada, Brooke, Tofiloski, Voll & Stede, 2011). This assumption claims that all negation cues turn the polarity value tokens under their scope to their opposite valence. This negative effect occurs regardless of the particular type of negation cue involved; for instance, if the polarity value of a positive word is 5, the same negated word’s polarity becomes -5 no matter which negator (i.e. “not,“ “never,“ etc.) causes such a reversal. It is worth mentioning that Taboada et al. (2011), recognizing the limitations of this approach, propose a new method called polarity shift (or “shift negation“) that shifts the negated word’s polarity for a constant amount (“4“ in their solution) across negation cases. Althogh an important improvement, Taboada et al.'s solution still does not take into account the diverse nature of negators because it uses a universal valence across all types of negation cues.

This standard reversing assumption is challenged by Zhu and colleagues (2014) who, after analyzing manually annotated sentiment scores in the Stanford Sentiment Treebank corpus, discovered that the phenomenon of negation introduces complexity and nuance that severely limits the conceptualization of said reversing assumption. Figure 1 visualizes the negation phenomena operationalized in that corpus. Each dot in the figure represents the sentiment score of a negated phrase. Values in the *x* axis display negation scope scores (represented as $s(\vec{w})$; an example of negated scope: "very good" in "isn't very good"), while the *y* axis shows scores of each negated phrase ($s(w_n,\vec{w})$, example: "isn't very good,"). The red diagonal line corresponds to sentiment scores for the standard reversing assumption:

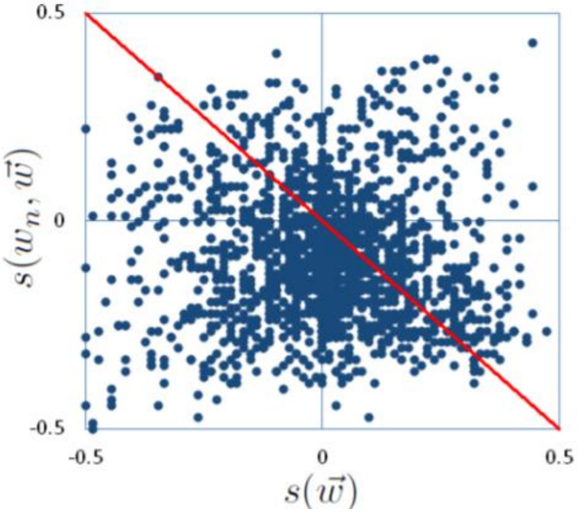
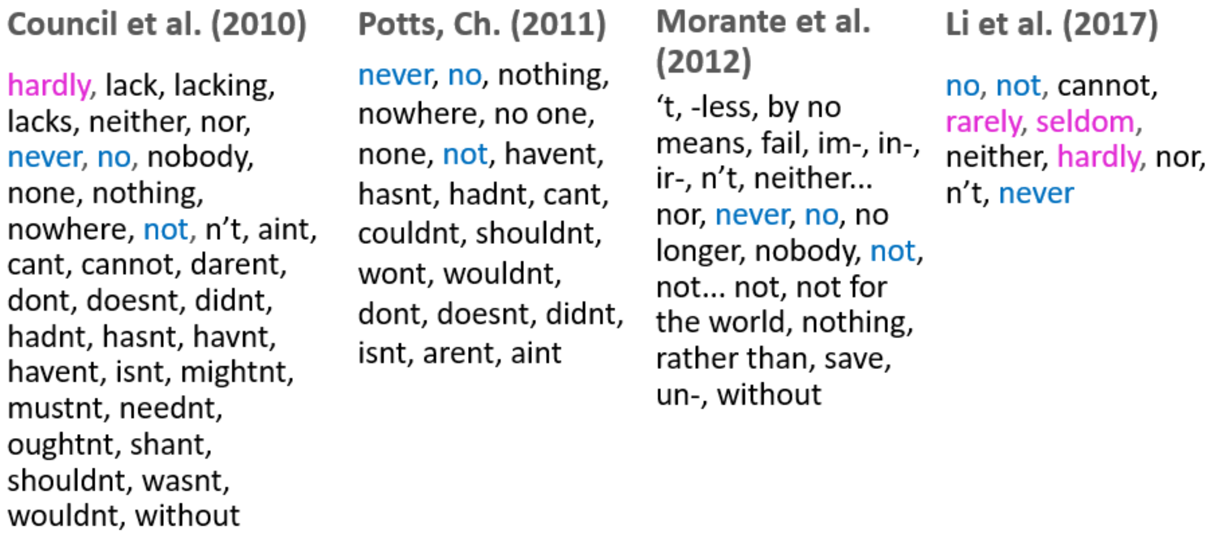


Figure 1: Negation effect of common negators in the Stanford Sentiment Treebank (Zhu et al., 2014, p. 304). Each dot represents a negated text score (based on human annotation values). The red diagonal line corresponds to values following the standard reversing assumption.

As we can see, the majority of sentiment score values assigned by humans are dispersed out in the classification space and away from those corresponding to standard polarity reversal. Indeed, humans assign different polarity intensities to the effect of distinctive types of negators. Based on this finding, these authors designed the discriminative approach to negation modeling that was used for their winning solution in the 2014 SemEval's sentiment analysis in Twitter task (task 9, mentioned above). In that new approach, instead of attaching a uniform negative tag to every token under the scope of every negator, the authors used the specific negation cue in place in each particular context. For example, the phrase "this is never acceptable" will be pre-processed as "acceptable_beNever" instead of a more generic tag such as "acceptable_NOT" (Zhu et al., 2014, August; further discussion in the Literature Review chapter). Indeed, expanding the types of operationalization of the negation phenomenon proves to be effective for improving negation prediction.

Disagreement in negation cues across lexica. This complexity in the semantic import of negators is also reflected by the fact that the list of negation cues used for modeling varies across research reports, as follows: Councill et al. report using 35 negation cues (2010, July); Potts suggests 21 (2011, November 8-9); Morante, Schrauwen, and Daelemans compile 22 (2011, May); and Li et al. employ only 10 (2017). Some of these lists include particles such as prefixes (e.g. "im-"), suffixes (like "-less"), multiple expression words, and contractions; while in other lists only single negation words are considered. Furthermore, Li et al. incorporate tokens "rarely," "seldom," and "hardly," which (as demonstrated by this report's findings) do not always convey full negative import. Only the words “no,“ “never,“ and “not“ overlap across the abovementioned negation lexica. Table 1 offers the full list of these negation cues:



Council et al. (2010)	Potts, Ch. (2011)	Morante et al. (2012)	Li et al. (2017)
hardly, lack, lacking, lacks, neither, nor, never, no, nobody, none, nothing, nowhere, not, n’t, aint, cant, cannot, darent, dont, doesnt, didnt, hadnt, hasnt, havnt, havent, isnt, mightnt, mustnt, neednt, oughtnt, shant, shouldnt, wasnt, wouldnt, without	never, no, nothing, nowhere, no one, none, not, havent, hasnt, hadnt, cant, couldnt, shouldnt, wont, wouldnt, dont, doesnt, didnt, isnt, arent, aint	‘t, -less, by no means, fail, im-, in-, ir-, n’t, neither... nor, never, no, no longer, nobody, not, not... not, not for the world, nothing, rather than, save, un-, without	no, not, cannot, rarely, seldom, neither, hardly, nor, n’t, never

Table 1. Examples of list of negation cues as reported negation lexica.
Words in blue overlap across lists. Words in pink are part of the focus of the present research study

These findings demonstrate that there is a window of opportunity for research that focuses on the nature and types of occurrences of negation cues themselves (besides negation scope) as a potential way to improve negation detection. The present research will take such an approach.

Negated context modeling issues for Twitter data. Additionally, and related to scope detection in social media messages, the ungrammatical nature of utterances such as tweets makes it difficult to rely on punctuation marks for scope modeling (as required by negated context modeling). Consider the following tweet:

> “SHUT IT DOWN. NOT MY CIRCUS NOT MY CLOWN don't give in unless we get what we want #TrumpShutdown…https://t.co/m191IgYI4e“
> (tweet id_str: 954533178638815238)

We can identify three negated phrases in this tweet: 1. “not my circus;“ 2. “not my clown;“ and, 3. “don’t give in.“ We also notice that there are no punctuation marks separting them out. Indeed, the only punctuation mark present in the tweet (a period after “shut it down“) is not related to any of those three negated phrases. A negated contexts solution will fail to identify the end of each scope because it depends on finding a final punctuation mark (presumably after “circus“ for the first scope, after “clown“ after the second, etc.)

Tweets like this example are prototypical of the casual writing style widely present in Twitter data. Due to the limitations of negated contexts in handling this phenomenon, an approach that considers structural relationships among words (such as dependency graph relations as introduced by Lapponi, 2012) helps resolve negation scope for this particular type of corpora. Such an approach is also taken by the present study.

Purpose of the Study

This study investigates the role of approximate negators as full reversal valence shifters or prototype negators and how to model their automatic detection using natural language processing techniques. It is important to highlight that this research investigates approximate negators behaving as full negators (i.e. negating a given scope), regardless the specific intensity or polarity valence conveyed by such a negator. Recognizing the complexity of negation operationalization introduced by the variety of negative valences that each negation cue conveys has demonstrated the need for this research, and also opened the opportunity to investigate non-prototype negation cues, such as the adverb’s object of this inquiry (namely “barely,” “hardly,” “rarely,” “scarcely,” and “seldom”). However, an exclusive focus on the nuances in types of negative import among these words is beyond the scope of this research endeavor.

Two components are part of a negation model for automatic detection: (i) defining the negation cue, and (ii) deciding on the effect of the cue over neighboring tokens (Wiegand, Balahur, Roth, Klakow, & Montoya, 2010; Morante & Blanco, 2012). In the particular case of defining the negation cue, efforts such as the tasks included in the abovementioned SEM 2012 Conference always assume the presence of a prototypical set of tokens that clearly operationalize full negation, such as “not,” “none,” or “never.” However, linguistic studies also discuss a differentiated set of words called “approximate negators” (Pullum & Huddleston, 2002) which, under certain circumstances, actually behave as full negation cues, such as “not” or “no.” Consider the following example (offered by Pullum & Huddleston):

“Hardly any of them complained, did they?” (p.820)

In this case, the structure of the sentence includes what is called a reversed polarity tag, “did they?” which possesses a positive polarity. This tag expresses a polarity that is invariably the opposite of that observed in the preceding sentence; their role is to seek confirmation from the listener. However, if the polarity tag of this sentence is positive, and its role is to confirm an opposite polarity shown by the preceding sentence, then that preceding sentence should have a negative polarity. In spite of that, we do not see a prototypical negation cue present, such as “not” or “none.” What we do see is the presence of the word “hardly.” As the linguistic literature explains, “hardly” is part of the set of approximate negators which, under some circumstances (such as positioning themselves at the beginning of the sentence) actually behave as full or prototypical negation cues, such as “not” or “never.”

Words such as “hardly” have also been investigated in some of the existing natural language processing literature as belonging to the set of contextual valence shifters, or tokens that affect the base polarity valence of neighboring words in positive or negative directions (Polanyi & Zaenen, 2004). However, according to a substantial subset of the literature, no particular model of these approximate negators behaving as full or prototype negators has been formulated, even in spite of the fact that linguistic theory has repeatedly investigated this phenomenon. This study attempts to make a contribution in this direction.

An important component of modeling negation is to accurately identify and label its scope. To prepare for doing so, this study discusses the latest reported advances in automatic detection of negated scope and event. Finally, the research design aims to answer the postulated inquiries by taking advantage of content analysis for analytical purposes through identifying the ways in which negation occurs in the data, and also endeavors to curate a corpus which will become the gold standard for machine learning experiments. These experiments aim to automatically detect negation cues and their scope when approximate negators acting as prototypical negators are present.

Preliminary Study

In order to explore this potentially significant contribution to negation modeling for computational linguistics, a preliminary study was performed using Twitter data with encouraging results. The total sample used for the study consisted of 2,800 tweets containing 14 different tokens, including the 7 approximate negators discussed in the pragmatic linguistic literature, which is summarized in the literature review that follows. This sample consisted of sub-samples of 200 tweets per token, which was further broken down into two subsets: a set of 100 tweets showing the token by itself in the tweet (called non-negated tweet subset), and another set of 100 tweets with the token accompanied by the negation cue “not” (or negated tweet subset). In the case of the approximate negators shown in non-negated tweet subsets, the rates of their operationalization as full or prototypical negation cues ranged between 12% and 44% within their 100 tweet sets (see the discussion “Conclusions from the preliminary study” in Chapter 2). Hence, it is apparent that the usage of English language on Twitter could be a good fit for examining the manifestation of this phenomenon.

Problem Statement

As discussed, negation modeling needs to be optimized in order to improve overall sentiment analysis performance. In particular, the study of nuances in types of negation cues has proven to be effective for this optimization task (see the discussion on SemEval shared tasks results; also Zhu el al., 2014, August). The analysis of words other than the standard negative items (such as “no” or “never,” preeminently used to identify negation), offers a window of opportunity for discovering those nuances in negation operationalization.

Research Questions

This research project hypothesizes that this specific subgroup of valence shifters, the approximate negators, operationalize full or absolute negation on particular occasions, and that modeling such negation behavior, along with the scope of its influence, has the potential to improve automatic detection of negation by machines. More specifically, the study aims to answer the following research question and sub-questions:

- Research question: How do approximate negators operationalize reversal shifting (i.e. prototypical or full negation) in Twitter utterances (tweets) in the context of improving automatic detection of negation in natural language processing?
    - *Sub-question 1*: In which ways do approximate negators reoccur when behaving as reversal shifters (prototype negators or negation cues) in tweets?
    - *Sub-question 2*: How do we automatically detect approximate negators behaving as reversal valence shifters (or prototype negators) in tweets?

As already emphasized, the study of these non-prototype words will involve investigating their negation behavior regardless of intensity or nuances in their particular negative polarity effect. In that sense, they all will be called “full reversal shifters” or “prototype negators” without further scrutiny on the intensity of their negative import that could distinguish or differentiate them from each other. Such an inquiry could be the aim of future studies.

Summary of Research Methodology

The basic research methodology consisted of content analysis followed by machine learning experiments. For content analysis, human annotators went over a dataset of tweets in order to: (i) identify approximate negators (more precisely adverbs “barely,” “hardly,” “rarely,” scarcely,” and “seldom”) acting as negation cues; and (ii) determine the scope of those cues over other tokens. While identifying negation cues, human annotators also marked reoccurrences of those words that signal negation. Those reoccurrences typically consisted of combinations of words (i.e. “barely anything” as negation cue) or syntactic traits (such as reversing auxiliary verb and subject in a sentence). In the machine learning experiments phase, those reoccurrences were used to define features for the classifier.

Limitations of the Study

The two main limitations of this study are: (i) the focus on the use of negation in the English language as operationalized in social media (Twitter); (ii) the use of a corpus based on highly noisy and ungrammatical text data (tweet-style utterances). Regarding the first limitation, the findings in this report apply to styles of English language employed by Twitter users in their everyday communications; thus, their generalization to other languages and/or other styles within English is limited. With respect to Twitter data, the presence of noisy data (such as tweets generated automatically by robots, or “bots”); grammatical inconsistencies such as misspellings and typos; and spurious sentential items (such as emoticons, URLs, and similar tokens), can jeopardize the validity of results if not handled properly in data pre-processing tasks. For this study, this later limitation was handled by identifying bot-generated tweets and also by using state-of-the-art natural language preprocessing tools, as explained in Chapter 3.

Thesis Structure

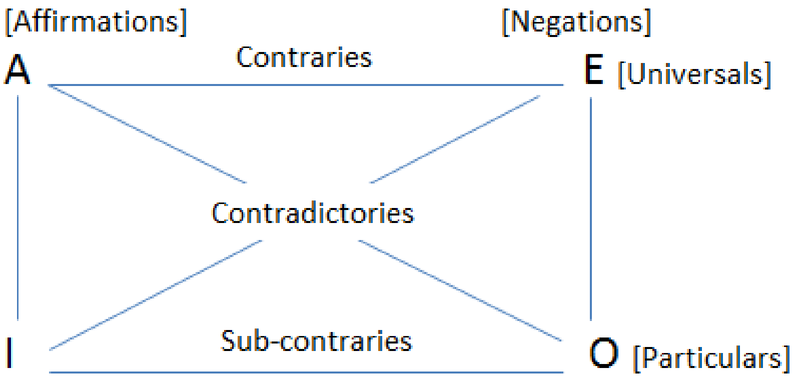
The document is divided into four chapters. After the present introduction, chapter 2 discusses previous efforts to understand this phenomenon as reported by the scholarly literature on negation and computational linguistics. Chapter 3 describes the research design (qualitative and quantitative methods) and other methodological strategies that support this inquiry. Research results are reported in Chapter 4, with detailed information on both human observation notes and machine learning findings. Finally, the last chapter (5) offers reflections on limitations of reported results along with future lines of work.

Chapter 2: Literature Review of Negation in Linguistic and Computational Linguistic Studies

Negation in Linguistics

Researchers recognize that negation as a phenomenon can be an elusive subject of study (Harabagiu, Hicki & Lacatusu, 2006, July; Wiegand et al., 2010). In this section, major contributions to the literature on negation (instrumental to this research report), from both a linguistic and computing modeling approach are reviewed. It is worth reminding the reader that this research focuses on the English language; consequently, this literature review will focus on studies in negation as they pertain to the phenomenon within the English language only.

In “A natural history of negation,” Horn (1989) presents a comprehensive history of the linguistics and semantics of negation from a comparative linguistic approach. He explains that the study of the phenomenon of negation started as a philosophical problem in arguments from as far back as those between Aristotle and the Stoics. In the case of Aristotelian logic, negation is explained in the context of the square of oppositions, where affirmative and negative categorical statements occupy opposite sides of a square:



A: Every man is white
I: Some man is white, some men are white
E: No man is white
O: Not every man is white, some men are not white

Figure 2: Square of oppositions in Aristotelian logic (Horn, 1989, p. 12)

In this square, we can see that quantitative categorization descends vertically from a universal or absolute state of things towards a particular or individual one, while mutually exclusive affirmative and negative statements belong in the horizontal columns (left and right). Quantitatively, negation is defined as denying a state of things in either universal or absolute terms through the use of contrary statements, such as “every man is white” and “no man is white.” Negation can also be defined through particular or individual situations by sub-contraries, i.e. “some man is white, some men are white” versus “not every man is white, some men are not white.” Contradictories, which travel in diagonal directions, occur whenever the negation crosses from *a particular* level to *the universal* levels (or vice-versa), as in the case of “some man is white, some men are white” and its contradiction, “no man is white.”

As already mentioned, the concept of contraries and contradictories is based on quantitative aspects of negation. Additionally, the mutually exclusive relationships between the statements in the figure rely on the assumption that negation states can either be true or false. Once denied, the state of things becomes false; after affirmed, the state of those same things turns true. In other words, in a given state of things, only one member of the pair (either the affirmation or the negation, the universal or the particular) can be true, and the other will necessarily be false. “No man is white” negates “every man is white” as its contrary because one must be true and the other false, while the same applies to “not every man is white, some men are not white” as the negative contradiction of “every man is white.” This assumption is called the Law of the Excluded Middle: things can only be affirmed or negated or, what is the same can be true or false, with no middle or intermediary ontological status. Such an assumption, along with the presupposition that it is impossible to be and not to be at the same time (a condition called the Law of Contradiction), is at the foundation of the Aristotelian logic for explaining negation, among other states of things.

Aristotelian logic works within the boundaries of terms, i.e. words that act as subjects around which other words function as predicated by affirming or denying states of matter. The Stoics came to define negation beyond limits of terms by creating propositional logic, which involves the analysis of entire sentences as units and potential interactions among them. Thanks to propositional logic, hypothetical statements such as “if P then Q,” or disjunctive categories, like “P or Q,” can be explained. However, although the scope of their theoretical framework allowed a broader understanding of the phenomenon, the mutually exclusive categories of false versus true still remain at the foundation of the traditional logic approach to the conceptualization of negation.

Pragmatic Approach to the Study of Negation

As explained in the Introduction, this research thesis seeks to make a contribution to the field of automatic detection of negation, which will eventually improve sentiment analysis by detecting negative sentiment more effectively. Indeed, the use of natural language processing to explain negation phenomena is clearly stated in the research question: "How do approximate negators operationalize reversal shifting (i.e. prototype or full) negation in Twitter utterances (tweets) in the context of improving automatic detection of negation in natural language processing?" Precisely for this reason, the research design (discussed in Chapter 3) involves natural language processing and machine learning to manipulate text and make predictions. At the same time, and by definition, natural language processing works with negative expressions as they are operationalized within language that is in use. This fact should not be overlooked when seeking to build a theoretical framework that can explain negation in the context of automatic detection by machines. Consequently, this research document approaches the analysis of utterances from a pragmatic linguistics standpoint, in line with previous approaches adopted by computer linguistics researchers such as Councill et al. (2010, July) and Morante and Blanco (2012).

The very study of negation within pragmatic linguistics actually began because scholars identified a gap between the functionality of negation in everyday speech (called natural language in computational linguistics) and explanations of the phenomena offered by linguistical logic frameworks such as those of Aristotle and the Stoics (Horn & Kato, 2000). Let's take for example the use of double negation. From a standpoint of logic, two negative words within the same expression should cancel each other out, returning positive semantic meaning: “it is **not un**necessary[2]” actually means that something is necessary. Although this principle can be observed in natural language, it is also a fact that double negation in everyday speech sometimes represents the opposite: the need to emphasize negative meaning, as in “I ca**n’t** get **no** satisfaction.” This hypernegation phenomenon, in this case also called negative concord, has been widely addressed in pragmatic linguistics (Horn, 2011). As Jaspersen states: “when logicians insist that ‘two negatives make an affirmative,’ their rule is not corroborated by actual usage in most languages” (1917, p. 62). But mutual cancellation and negative concord are just two ways to interpret a sentence with double negation. In the expression “she is **not un**happy,” there is double negation with no straightforward negative emphasis, and, interestingly enough, there is no direct negative cancellation either. Actually, “she is not unhappy” does not automatically rephrase into “she is happy;” it may also mean “she is not fully unhappy, but she is not happy either” (Blanco & Moldovan, 2011, March, p. 228).

Negative concord is just one instance among multiple operationalizations of negation that makes this phenomenon complex beyond conventional terms of logic. Other equally complex manifestations of negation are content negators or verbs that negate meaning, such as “lacked” or “denied” (Councill et al., 2010, July); the use of quantifiers like “few” or “little;” and anaphora, or whenever one negative connotation of a particular expression depends on another one (Blanco & Moldovan, 2011, March). It is precisely this complexity, along with the empirical approach of natural language

[2] Throughout the document, negations will be highlighted in bold for emphasis or clarity whenever considered useful for the reader

processing studies, which makes an inquiry into negation using mere logical linguistics insufficient.

Main Contributions of Pragmatic Linguistics to the Understanding of Negation

In his foundational work “Negation in English and other languages,” Otto Jespersen (1917) opens the study of negation from a pragmatic linguistic approach by looking for occurrences in corpora. Jespersen explains that the meaning of negation is related not only to logical principles, but also to the way in which words interact with each other, as well as to an utterer’s implicit expectations. For example, “not good” literally means “different from good,” which potentially involves something which is less than good but could also mean superior to good. However, in real usage, utterers use this expression to imply “inferior” more often than they do to invoke “excellent.” The negation in the sentence “he doesn’t spend $200 a year” implies that he spends less than $200, but the use of “not” in “Rome was not built in a day” represents the opposite (Rome certainly took more than a day to build). We can plainly see, then, that negation behaves differently according to contextual factors (such as usage) as well as according to interactions between particular words.

In an attempt to articulate this phenomenon, Jaspersen applies the notions of contrary and contradictory (offered by logicians) to the empirical elements he finds in natural language. While for him the notion of “contrary” still encapsulates two opposite entities (like “hot” and “icy” or “all” and “nothing”), Jaspersen also proposes a wide range of statuses between those extreme ends, such as “hot (sweltering) - warm - tepid - lukewarm - mild - fresh - cool - chilly - cold - frosty - icy” (Jespersen, 1917, p. 85). In the case of negation, Jespersen establishes three stages: absolute positive (such as “all”), intermediary (“something”), and absolute negative (“nothing”). Different usage between quantifiers and negators allow speakers to negate in a wide range of intensity, such as “not all girls” with a closer meaning to “some girls.” On the contrary, the notion of contradictory negators indicates absolute opposition, “A and not-A comprising everything in existence” (p. 80) or, in other words, there is nothing between A and not-A, as in “ever” and “never,” as logicians will interpret it.

Around half a century later, Edward Klima also emphasized the non-logic and contextual aspects of negation, to the extent of declaring that “labels like *negative* have no meaning above and beyond their grammatical function of specifying a structural position and some difference from other symbols” (Klima, 1964, p 247; emphasis in original). According to this interpretation, text tokens cannot be assumed as negative by themselves but only as relative to their role within a particular sentence, functioning as a structure with other tokens. Here is when the notion of scope of negation becomes essential to understanding this phenomenon, since Klima’s approach assumes a transformational grammar framework. Negation is found to affect most tokens of the structure it belongs to, with negation cues having great flexibility to attach to different parts of the sentence or even parts of words within the sentence (in the form of affixes). For example, the sentence “I would force her to marry no one” is equivalent to “I would**n’t** force her to marry **any**one,” with negative cues acquiring the forms of “no,” “n’t,” and “any.” However, in the two equivalent cases: “I would force her **not** to marry **any**one” and “I would**n’t** force her to marry **any**one,” the former holds a positive meaning and the latter a negative one in spite of the fact that both show two negation cues (Klima, 1964, p. 303). This phenomenon occurs due to the highly contextual nature of negation, which only makes it possible to analyze it within its local semantic structure at the sentential level. Additionally, negation is multimorphic: although the negation cue “not” scopes over most tokens within a sentence, “because that element is mobile and capable of fusing with other elements… its ultimate position and form have great latitude” (p. 316). The work of Klima was pioneer in the study of tokens whose co-occurrence signal negation, and the lexicon he created was used later by natural language processing researchers in the investigation of this phenomenon (Councill et al., 2010, July).

More recently, Gunnel Tottie (1991) contributes to the pragmatic linguistic analysis of negation from a communicational point of view, beyond the model of mere opposites (contraries and contradictories). After years of studying the London-Lund and Lancaster-Oslo/Bergen corpora, Tottie argued that negation operationalizes in communication in two ways: rejections and denials. Rejections involve declining explicit proposals, while denials refute assertions or situations. In turn, denials break down into two types: (i) explicit, or those related to unambiguous assertions or situations; and (ii) implicit denials where expectations are not met or contextual conditions make the utterer refute tacit elements. Compare these examples taken from the Svartvik & Quirk (1980) corpora (as referred to by Tottie, 1991, on page 28):

- explicit denial: “well, he didn’t say that”
- implicit denial: “I just boiled some water for coffee cos I haven't had time for tea” (the utterer responds to some implicit expectation stating that he should have had coffee or tea already, such as in the case of the time after breakfast)

Tottie explains that implicit denials occur often in written communication, such as when a writer discusses a topic assuming a specific shared background with the reader, while rejections and explicit denials are more suited to dialogical contexts since an utterance should be presented to the receiver for him or her to reject or explicitly deny.

A final linguistic framework for negation that should be mentioned in this literature review is offered by Pullum and Huddleston (2002) in "The Cambridge Grammar of English Language" chapter dedicated to negation, since it has been employed as a linguistic framework for the SEM 2012 Conference tasks in modeling the scope and focus of negation, as we will review below.

Pullum and Huddleston define negation as negative polarity, which is marked by applying specific words ("not," "no," "never," and the like) or affixes (such as "n't" or the prefix "un-") to sentences and words holding positive polarity. More specifically, these authors mention four tests that can be conducted on a sentence to decide if its polarity is negative:

- Clause continuation with "not even" and a complement or adjunct, as in "he didn't read it, not even the abstract." Since "not even" is not acceptable following a positive clause, every clause displaying such an expression should contain a negative polarity.
- Use of connective adjuncts "neither" or "nor" following a sentence define that sentence as negative (as in "He didn't read it; neither/nor did I"). Conversely, positive polarity clauses are followed by "so" ("Ed read it; so did I").
- The presence of reversed polarity tags (or reduced interrogative clauses), used when looking for confirmation from the recipient of the utterance. Since

the polarity of these tags is reversed, a positive tag will indicate a preceding sentence with a negative polarity (as in “Ed didn’t read it, did he?”).

- Occurrence of subject-auxiliary inversion with prenuclear constituents: since negation always falls on a verb (Jespersen, 1917), an auxiliary is needed when negation occurs before the subject (for example, “Not once did Ed read it”).

These four tests of polarity can act as rules of thumb when deciding whether a statement is negative or not (and they were employed as such in this research, as explained in Chapter 3). We can see that these are basically syntactic rules, revolving around the presence of expressions in particular positions within a sentence (when analyzing the scope of negation, however, Pullum and Huddleston point out that the concept of scope should rather be defined in semantic terms, so a semantic analysis is required).

Following such an assumption, these authors discuss a specific test to define the scope of negation, which consists of eliminating the negation cue and analyzing the semantic effect of its removal. For example, “Liz didn’t delete the backup file” turns into “Liz deleted the backup file.” Now, in order to identify the scope of the negative, we analyze the three elements “Liz,” “deleted,” and “the backup file,” and discover that they necessarily cooperate with the meaning of the whole sentence (i.e. there was a deletion performed, which Liz did, and she did it on the backup file). In other words, the three elements must be true in order for the sentence to be true or to be considered a positive statement (which is the equivalent in linguistic logic). Conversely, making any of the three elements false turns the whole sentence into a false statement, or, into a negation of a true state of things. If Liz didn’t perform the action (because somebody else did it), then “Liz didn’t delete the backup file” is correct. But if the backup file was not affected, or if there was no action of deletion, the negative sentence is still correct. Consequently, the negation effect of “n’t” overwrites all three elements of the sentence. The semantic scope of negation is then defined as the specific elements of a sentence that are affected by a particular negation cue by turning their polarity from positive or true to negative or false. Interestingly enough, Pullum and Huddleston represent the semantic scope of negation within a framework of logical linguistics, i.e., going back to affirmation in terms of true states and negation in terms of false ones.

Although the concept of scope is primarily semantic, forms of syntactic scope should also be considered. Pullum and Huddleston explain that syntactic scope usually overlaps the semantic scope in terms of the number of tokens affected and their relationships with one another. Syntactic scope is also linear and sequential: scopes appear one after the other and, in the case of overlapping scopes, the first one also involves the second.

The pragmatic tradition of negation research contributes elements which are highly relevant to the understanding of this phenomenon in the context of natural language processing. The wide range of ways in which negation tokens affect other words (Jespersen’s contraries); the analysis of scope and the grammatically flexible nature of negation cues (Klima’s transformational grammar approach); the semantic motivation of the utterer for using negation (Tottie’s concepts of rejection and denials); and the rules for deciding if a sentence holds negative polarity or not (Pullum and Huddleston), all contribute to later formulations of negation in machine learning modeling. As an example, the concept of valence shifters (as formulated by Polanyi & Zaenen, 2004), incorporates a definition of negation that borrows elements from the linguistic tradition of mutually exclusive contraries or opposites, but adds intermediary valences between those opposite ends. Additionally, the formulation of valence shifters gathers elements from a particular operationalization of negation that has also been reviewed in the literature of pragmatic linguistics: the incomplete or approximate negators. The next section introduces this concept and reviews the literature related to it, as it will become the central element of this research report.

Approximate Negators

So far, we have discussed negation as operationalized by specific words or negation cues that are clear markers of negated statuses. Words such as “no,” “not,” “never,” or “any,” undoubtedly bring about operationalizations of negation. However, some other specific words which do not belong to the negation list can acquire “negative import” (as Jespersen termed it) within particular contexts or when following specific behaviors. Although scholars defined these words differently in the early literature, more recent linguistic theory (Pullum & Huddleston, 2002) calls them “approximate negators.” The next section attempts to elucidate the particular way in which negation operationalizes in the English language with approximate negators.

As Jespersen (1917) explained, negation goes far beyond mutually exclusive opposite states. It was mentioned above that his initial formulation included one intermediary state, “something,” standing between “all” and “nothing.” Additionally, he also discusses what he calls “incomplete negators,” or words that represent intermediary states between affirmation and negation. The list of incomplete negators includes adverbs such as “hardly” and “scarcely” and quantifiers like “little” and “few.” The following is a summary of Jespersen’s discussion of the role played by these in negation (as elaborated in 1917, p 39-42):

- “Hardly,” meaning “with hardness, i.e. with difficulty,” represents “almost not” and is usually strengthened using the expression “at all.” Jespersen notes that “hardly” follows the way other negatives occur when being placed before the negated element, such as in “I hardly know.”

- “Scarcely” represents “not quite” and it’s also defined as a “restricted negative.” Its negation import is represented by the fact that sentences such as “scarcely any” or “scarcely ever” are more frequently used than their equivalents “almost no” or “almost never.”

- “Little” and “few” may also show negative import, as when they are used with “yet” (for example, “I have yet seen little of Florence”). The combination of these words with “no” or “none” also reveals their semantic closeness to negation, as in “there is little or no danger.”

    - “Little,” in particular, is commonly placed before the verb, as negation cues often are (“they little think what mischief is at hand”). One exception of this negative connotation is given by the sentence “love me little and love me long,” where “little” holds positive meaning.

    - “A little” and “a few” have a rather positive import, and should be distinguished from the negation set. For instance, “little” means “less than you would expect” while “a little” represents “more than you would expect.”

        - Particularly in American English, “a little” and “a few” can be found along with “quite,” reinforcing its positive connotation, such as in “quite a little,” meaning “a good deal” or “quite a few,” representing “a good many.”

Klima (1964) also discussed incomplete negation when making a case around the complex nature of negation as a whole. Different aspects of negation can indeed be found in a wide range of sentences such as in the following (p. 249, italics in original):

The students did *not* believe that it had happened (1a)
The students *never* believed that it had happened (1b)
The students *hardly* believed that it had happened (1c)
The students *rarely* believed that it had happened (1d)
None of the students believed that it had happened (1e)
Few students believed that it had happened (1f)
The students were *unable* to believe that it had happened (1g)
The students were *too* intelligent to believe that it had happened (1h)
The students *doubted* that it had happened (1i)

Klima explains that only (1a) and (1b) can strictly be considered negation sentences because they show clear negation cues ("not" and "never"). The rest of the sentences in the list "contain incomplete, special or inherent negatives" (p. 250). "Unable" in (1g) actually shows a negation form, by means of the prefix "un-" negating the verb "able;" while "none" in (1e) is considered a negative form of "one." In (1i) we found another example of content negators, or verbs with negative connotation with the use of the verb "doubt." Additionally, not only does the semantical interaction of words in these sentences carry out negative meaning, the words themselves have diverse structural roles. "Hardly" (1c) and "rarely" (1d) are adverbs, while "few" (1f) is a quantitative adjective. Finally, the use of "too" in "too intelligent to believe" (1h) has a negative connotation that contradicts its original, commonly used positive valence.

According to Klima, the multiple and diverse nature of these incomplete negations prove that the very structure of language (rather than the presence of specific negation cues or symbols), plays a primary role in determining how negation operationalizes, to the extent that we should ask ourselves: "is it the case that a single symbol accounts for certain linguistic facts at the very places where negativeness is intuited?" (p. 250).

More recently, Pullum and Huddleston (2002), drew a newer formulation of approximate negators within the framework of linguistic theory. This formulation has proven to be instrumental to natural language processing, especially for defining conceptual elements in the SEM 2012 Conference tasks on modeling the scope and focus of negation (Morante & Blanco, 2012). Pullum and Huddleston define approximate negators as “imprecise quantifiers” with “non-zero implicature” (p. 816). For example, let’s say that the sentence “Many have resigned” has polarity 1 (positive), and “None have resigned” has 0 (negative). Related to these expressions, we have sentences such as “A few have resigned” or “Few have resigned” (using the approximate negators “few”), covering a range of valences that go from close to 1 (or absolute positive), down to close to 0 (or absolute negation), but without reaching that absolute or full negation. This helps to illuminate why the phenomenon is named non-zero implicature or non-absolute inferred negative meaning in the utterance.

Syntactic test for negative polarity. An element which bears highlighting in Pullum and Huddleston’s discussion of approximate negators is the fact that, according to these authors, such negators actually turn themselves into absolute negators under certain conditions or occurring in specific ways, like being positioned early in a sentence (as in “**Few** of the boys had shown any interest in the proposal”). In other words, they acquire a “zero implicature.”

In those cases, the authors advise conducting one of four possible negative polarity tests in order to decide if the approximate negator acts as an absolute negator or negation cue. As mentioned above, the four tests for negative polarity are as follows: (i) adding a continuation led by “not even;” (ii) appending the connective adjuncts "neither" or "nor;" (iii) attaching a reverse (positive) polarity tag or expression that holds the opposite polarity of the sentence, and that is used for confirmation ("Ed didn't read it, did he?"); and (iv) identifying the presence of subject-auxiliary inversion with prenuclear constituents ("Not once did Ed read it"). According to Pullum and Huddleston, then, if a sentence holding an approximate negator passes one of these four tests, we are dealing with an absolute or full negation cue (as in the case of words such as "not" or "never"), with its corresponding scope and negated event.

The complete inventory of approximate negators comprises "few," "little" (determinatives), "rarely," "seldom," "barely," "hardly," and "scarcely" (adverbs). Although all seven words can potentially turn into absolute negation cues, Pullum and Huddleston point out that "rarely" and "seldom" tend to have weaker roles as negators.

The study of the approximate negators as a particular operationalization of negation could open a new view of this topic in the context of natural language processing, with potential impact on the improvement of automatic detection of negation. In fact, the concept of valence shifters incorporates the imprecise negative role that some tokens have when they affect neighboring words. In the next section, the study of negation as a natural language processing task is reviewed, with a focus on formulations involving approximate negators as a sub-group of valence shifters.

Defining Negation for Computational Linguistics

Computational linguistics researchers took advantage of findings from pragmatic linguistics in constructing a theoretical framework for their applied work. Drawing from authors involved in producing the literature discussed above, Morante and Sporleder

(2012) define negation as a way to grammatically express that an event, circumstance, or entity does not hold or exist, while Polanyi and Zaenen (2004) characterize it as changing (or shifting) the value of a proposition towards its opposite valence. As supported by the preceding literature review, this notion of shifting to the opposite is often behind negation modeling, whereas “negated polar expressions [are considered to be] unnegated polar expressions with the opposite polarity type” (Wiegand et al., 2010).

The role of contextual valence shifters. Polanyi and Zaenen (2004) classify negatives as valence shifters that either alter the polarity of another token by weakening or strengthening it, or even go so far as to turn it into the complete opposite. Following that definition, then, negations and their scope can work as valence shifters that may radically alter the polarity of words and, subsequently, the sentiment score of a sentence. Negative shifters sometimes act, in fact, as “reversal” shifters (Kennedy & Inkpen, 2006).

Polanyi and Zaenen offer a comprehensive taxonomy of contextual valence shifters. The authors explain how a particular type of valence shifters called “intensifiers” can strengthen or diminish the valence of the term under their influence, as in the case of “it is rather efficient” where “rather” weakens the positive valence of “efficient.” The authors suggest calculating the shifting value quantitatively by adding or subtracting a constant amount (or “a point”) to the base valence of the affected word, according to the enhancing or diminishing forces of the intensifier. For instance, in the example above, let’s say that the base value of “effective” is “2.” “Rather” will then diminish the base valence of “effective” by one point to “1,” while if we had the expression “deeply suspicious,” the enhancing intensifier “deeply” would add one point to the base value “2” of “suspicious,” consequently giving it a final point value of “3” (Polanyi and Zaenen, 2004).

Approximate negators as valence shifters. Valence shifters can indeed have dramatic effects over neighboring tokens in a sentence. Furthermore, a sub-group of valence shifters (such as “hardly” or “rarely”) also belong to the approximate negators group earlier identified by Jespersen (1917), Klima (1964), and Pullum and Huddleston (2002). As already mentioned, more recently Pullum and Huddleston include a total of seven words in the complete set of approximate negators: determinatives “few” and “little” and the adverbs “rarely,” “seldom,” “barely,” “hardly,” and “scarcely” (a total of seven words). These words may have an inexact negation effect on the tokens in their scope. Such an effect can range from bringing the word’s meaning to near complete or even full negation (such as in the case of “**Few** of them will survive,” which is close to “**None** of them will survive”), to triggering a lower shifting effect (as in the case of “Ed rarely leaves his house,” which implies that Ed positively leaves the house but only on special occasions). Following the formulation of context valence shifters, in the latter case these words act as diminishers, i.e. tokens that shift the valence of words to an incomplete turn only, lowering the semantic intensity of their valence (as in “hardly working,” for instance, where “hardly” does not negate the act of working but it sharply drops its base semantic value). In the former case, however, according to Pullum and Huddleston, approximate negators act as any other absolute negation cue (such as “no” or “not”). Ergo, they should be identified as such and their negation effect over other tokens in the sentence (i.e. scope) should also be identified.

Modeling approximate negators as part of reversal valence shifters in the context of natural language processing tasks. As discussed above, Polanyi and Zaenen (2004) identified contextual valence shifters as terms that change the positive or negative valence of another term. A negation cue is defined as “the most obvious shifter” (p. 4). Negation flips the positive valence of a word towards its negative, as caused by negation cues such as “not.” No formulation of approximate negators is made, although some of those words (such as “few” or “most”) are offered as examples of “intensifiers” or tokens that either weaken or strengthen the valence carried by neighboring words.

Although Polanyi and Zaenen made the case for applying context valence shifters to sentiment analysis, their contribution was only conceptual – “a proof of concept,” these authors claim – leaving the effectiveness of their model as an unanswered question. Later on, scholars Kennedy and Inkpen (2006) experimented with two applications of the context valence shifters model. The first experiment consists of counting words and adding their valences, while the second approach uses two SVM classifiers. Within this context, the authors define negation as shifting the sentiment of a term into its opposite. Intensifiers and diminishers are also considered. The results show that, indeed, modeling context valence shifters has a positive impact on sentiment analysis, with accuracy scores between 80 and 85.9%. However, the authors do not attempt to model the potential role of approximate negators as negation cues, nor do they include those words as such in the lexica they employ. The closest list of approximate negators they utilized are a set of words named “understatements and overstatements” (A. Kennedy, personal communication, June 6, 2015), but which do not include all seven words in the approximate negator set.

Two years later, Shaikh, Prendinger and Mitsuru (2007) released a report in which they introduced “SenseNet,” their linguistic tool for sentiment analysis. A fundamental element of the tool is an algorithm that recognizes contextual valence shifters and estimates the sentiment value of sentences that contain them. The authors perform semantic dependency parsing and analysis, incorporating an algorithm for valence shifters as a second step when estimating sentiment polarity. Although the authors do not explicitly mention the issue of approximate negators, they model three of the adverbs defined as such, namely “hardly,” “rarely,” and “seldom,” among others. Shaikh et al. call these approximate negators “exceptional adverbs” (p. 574) because of their potential imprecision, which they tackle by formulating a set of rules. However, these rules model the approximate negators’ effect in terms of bigrams, not in looking for a wider scope range. These authors do not consider exceptional adverbs as potential negation cues either, but rather as regular shifters.

Approximate negators acting as prototype negators in Twitter data: Conclusions from the preliminary study. The preliminary study performed on a dataset of 2,800 tweets as part of this research project demonstrated an abundance of the phenomenon of approximate negators acting as prototype negation cues on Twitter. The sample investigated the behavior of 14 valence shifter tokens (specifically downtoners taken from the literature), 7 of them also considered to be approximate negators. For each token, 200 tweets were collected: 100 where the token appeared by itself (or non-negated tweet subset), and 100 where the token showed up with the prototype negation cue “not” (negated tweet subset). The abovementioned syntactic test for negative polarity was performed on the non-negated tweet subset, with the following rate of occurrences passing the test: “barely” showing negation in 44% of cases, “hardly” in 25% cases, “seldom” did so in 20% of the cases, and “scarcely” and “rarely” in 12%, respectively. A couple of examples of these occurrences follow:

- Use of reverse polarity tag with “hardly” (bolding added):
  “RT @MarkNeary1: @sassyinthecityx "Robust" is another of those non- adjectives. **You'd hardly** have a flimsy action plan **would you**?”

- Use of subject-auxiliary inversion with “rarely” (bolds added):
  “Hey guys, so **rarely do u see** me post things that are negative or disheartening. I deal with them and post the... http://t.co/uW90tiR7ig”

Just in taking an initial look at these examples from the preliminary study, we can conclude that data collected from Twitter offers a suitable arena to analyze this particular operationalization of negation in English through approximate negators, which could ultimately lead to an improvement in its modeling and, it would follow, in the overall performance of automated sentiment analysis tools. Specifically, regarding sentiment analysis (and as explained above), Zhu, Kiritchenko and Mohammad (2014, August) reported that recent improvements made in negation modeling (for their sentiment detection solution for tweets) generated a significant impact on the solution’s F-score, raising it from 86.37 to 86.63. Indeed, whenever negative sentiment score raises, the overall performance of sentiment detection raises (Mohammad, Kiritchenko and Zhu, 2013, June).

However, identifying and analyzing a particular operationalization of negation cues is only one component of modeling negation. Another important element is defining the scope of negation or, in other words, determining which tokens are under the influence or effect of the negation cue. In the following section, I will review some of the existing literature on modeling negation cues and their scope.

Modeling Negation and its Scope in Computational Linguistics

Initial approaches to resolving negation for sentiment analysis used supervised machine-learning algorithms such as Support Vector Machines (SVM), as well as Naïve Bayes and Maximum Entropy classifiers. These techniques use a “bag of words,” i.e., they consider all words in a text regardless of grammar or relationship between one another. Although this approach to modeling negation can be considered satisfactory, it presents two problems: (i) it does not consider polarity, which prevents advanced sentiment analysis, and (ii) since the whole text is taken as a “bag of words,” words’ contexts are disregarded, leading to the improper modeling of the scope of negation (Wiegand et al., 2010).

The following is a summary of the most reviewed approaches to modeling scope for automatic detection of negation.

Use of Delimiters and Heuristic Rules

Jia, Yu, and Meng (2009, November) introduce the task of modeling negation scope when, while working on sentiment analysis for opinion retrieval, they try to resolve the issue of several negation cues occurring within a sentence. Specifically, when more than one negation word appears in a sentence, which part of the sentence does each one affect? Their approach first computes a candidate scope, and then proceeds to identify particular tokens (so-called “delimiters”) that help determine the real scope from the candidate scope by cutting back tokens that are actually not part of it. In a parse tree, the candidate scope, or “logical unit,” constitutes the group of descendent leaf nodes derived from a non-terminal node. When the logical unit contains a negation cue, that unit then becomes a candidate scope for such a cue; i.e. after defining the parse tree for a sentence, all descendent leaf nodes starting from the cue and towards the right hand-side of it are part of the candidate negation scope.

Now, the task is to find the real scope of negation within the candidate scope boundaries. For that, the authors discuss two strategies: (i) using a series of “delimiters,” and (ii) employing heuristic rules. Delimiters are tokens that will help eliminate words not affected by the negation cue. There are two types of delimiters. The authors simply call the first type “delimiter,” while Wiegand et al. (2010) expand it to “static delimiter.” These delimiters are words that clearly set boundaries for the negation scope. Examples of delimiters are tokens such as “when,” “whenever,” or “unless,” after which all other words are eliminated from the scope of negation. The second type of delimiter is called a “conditional delimiter,” because it will only eliminate words from the real scope of negation when particular conditions are met, such as what its part of speech tag is, whether or not it leads to an adjective clause or the location of the negation cue, etc. Examples of conditional delimiters are quotation marks and also words like “so,” “as,” “which,” “who,” etc.

The second type of approach for defining the real scope of negation consists of applying a set of rules affecting verbs, nouns, and adjectives with sentimental valence. Jia et al. call these rules “heuristic” because they do not rely upon a scope’s enclosing tokens, but rather on occurrences defined by sentimental terms. In the case of verbs, the heuristic occurrence dictates that the word immediately after a negated sentiment verb will become the negation's scope delimiter. Sentiment-loaded adjectives following a negated copula or verb also define the boundaries of negation scope, which is also the case whenever nouns are the object of a negated verb. Finally, when a negated verb has two objects, only the direct object remains within the scope, while the indirect object is excluded. The authors also offer cases where a negation cue does not have a particular scope. In some cases, a negation cue actually does not convey negation meaning (as in the expression "not to mention," for instance), in negative rhetorical questions (such as "who doesn't love cats?"), or the case of restricted comparative phrases such as "not better than." Finally, after the negation scope is found, the polarity of the part of the sentence within its boundaries should be reversed.

Syntax-based Approach

In their seminal report on sentiment classification, Pang et al. (2002, July) define the scope of negation based on syntax. The authors explored sentiment classification of movie reviews using three machine-learning methods: Naïve Bayes, Maximum Entropy, and Support Vector Machines. Pang et al. also defined negation as turning the meaning of an expression into the opposite. While explaining their decisions on feature selection, these authors decided on an "unconventional step" in attempting to model the contextual effect of negation. This approach constitutes an adaptation from an earlier technique (by Das & Chen, 2001, July) that decides the scope based on (i) negation cue and (ii) the first punctuation mark after such a negation cue. Pang et al. 's unconventional step is to create what we could call a “syntax frame” between the negation clue and its punctuation mark, and then add the unigram “NOT_” to every word within that syntax frame.

It is noteworthy that the winning research team of the SemEval 2013 followed Pang et al.’s approach for defining the scope of negation for their SVM classifier (Mohammad et al., 2013, June). They converted negation bigrams (i.e. “not good”) into unigrams by attaching the qualifier “NEG” to the negated unigram, regardless of the type of negation word, i.e. “not good” becomes “good_NEG.” This pre-processing work allowed these authors to attach polarity to the negation expressions as part of their negation model. The authors also added the negation qualifier “NEG” to polarity and emotion features, obtaining values such as “POLARITY_positive_NEG” whenever an expression with positive polarity was affected by a negative context (Mohammad et al., 2013, June).

Both Jia et al. and Pang et al. pioneered negation scope modeling. However, their solutions left two issues unresolved. First, they do not consider the scope of negation as potentially bi-directional, i.e. reaching out right and left from the negation cue. This limitation is due to the fact that these models search for words only to the right of this kind of cue, looking for either the next punctuation mark (Pang et al.) or a particular delimiter (Jia et al.). Second, they do not take into account the syntactic function of words as input in order to decide on the scope of negation. Further research was thus needed.

Automatic Detection of Scope Solution Using Dependency Graphs

Lapponi et al. (2012; 2012, December) introduced dependency graph information to create new features for negation scope modeling. These authors developed a set of features to train a Conditional Random Fields (CRF) classifier to label each token (called the vertex or node in dependency graph terminology) as in- or out-of-scope according to the type of syntactic relationship (or edge) it holds with the closest negation cue (another vertex or node) in a shared dependency graph.

A dependency syntax approach first finds syntactic heads or governing tokens from which edges or relationships are drawn towards other tokens left and right of the head. In the particular case of their SEM 2012 solution, Lapponi et al. (2012, December), offer the following example:

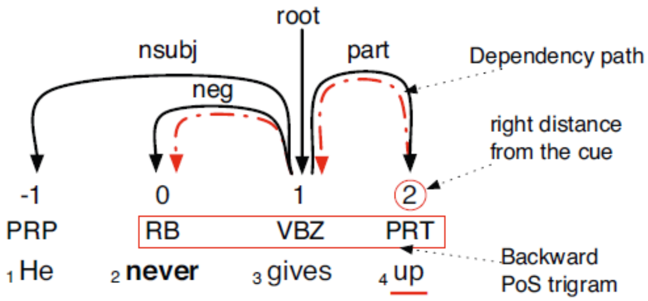


Figure 3: Dependency paths for a negated constituent by Lapponi et al. (2012, December, p. 689)

For this exemplary sentence, the authors' solution to resolve the scope of negation consists of first drawing dependency graphs (G and G') as sets of V vertices or nodes (i.e. tokens) that include directed edges (E), but also including bidirectional (i.e.

directed and reversed) edges (or E’). For the given example, the vertices and edges are determined as follows:

V = {1, 2, 3, 4}
E = {<3, 1>, <3, 2>, <3, 4>}
E’ = {<3, 1>, <3, 2>, <3, 4>, <1, 3>, <2, 3>, <4, 3>}

Generating two complementary dependency graphs:

G = {V, E}
G’ = {V, E’}

Hence, the scope solution extracts two complementary graph paths from each token to the sentence’s negation cue: (i) from graph G, the shortest path from the syntactic head of a negation cue to every token in the graph, and (ii) from graph G’, the shortest path from the negation cue to all tokens (head not considered), thus taking reversed edges into account. Thus, this solution then draws bi-directional relationships to the left and right of the head; whenever there is more than a negation cue, the authors use the nearest punctuation marks in order to decide the shortest path.

Furthermore, using bidirectional information, these authors draw a dependency path from each token to the nearest negation cue that also includes the particular edge direction (directed or reversed) required to reach that cue. Considering the example offered above, when positioned in index 4 for token-node “up,” the path to the nearest negation cue draws out as follows: (a) using reversed syntactic edge “part” for reaching the verb “gives” at index 3; and (b) taking the directed edge “neg” to get to the negation cue “never” at index 2. The way the direction of the edge is recorded is using downward arrows (“↓”) for directed edges and upward ones (“↑”) for reversed relationships. Thus, the dependency path from token-node “up” to negation cue “never” reads: “↑ *part* ↓ *neg*” (italics from the original; Lapponi et al., 2012, December).

Based on dependency graph information, Lapponi et al. introduce the following scope features: (i) directed dependency distance: or number of tokens from the present token-vertex to the token-cue as extracted from graph G; (ii) bidirectional dependency distance: the number of tokens linked by directed and reversed (i.e. bidirectional) edges, which are taken from graph G’; (iii) dependency path: as shown in the example above, and; (iv) lexicalized dependency path: which includes each token-node along with edges in the path.

In order to apply their model, the authors use the dependency parser Maltparser[3] for dependency labeling. The dependency parser finds the edges corresponding to graph G in the example, originating then further bi-directional edges from the position of the negation cue within that graph. It seems worthwhile to note that this solution offered the best performance for scope resolution in the 2012 SEM shared task.

Recent Use of Dependency Paths for Negation Modeling

Recently Cruz, Taboada, and Mitkov (2016) developed a solution for negation and speculation classification using dependency graph features borrowed from Lapponi et al. (2012, December). The solution proved again to be satisfactory, reaching an 84.07 F1 score in scope detection. The authors experimented using Naïve Bayes and SVM classifiers and obtained the best F1 scores with SVM using a cost-sensitive learning model (CS-SVM) to compensate for class imbalances in their product reviews database,

[3] http://www.maltparser.org/

the Simon Fraser University Review corpus. This corpus consists of 400 online reviews annotated with negation and speculation cues as well as their scopes, and it was developed as gold standard for automatic detection tasks. Particularly for the case of negation detection, their solution reached a F1 score of 89.64 using lemma and part of speech tags features.

Parse-and-paraphrase Solution

Liu and Seneff (2009) developed a linguistic solution that provides a hierarchical representation for a sentence, aiming to overcome the flat structure of strings. Their parser identifies hierarchical levels of semantic dependencies, encodes them, and then paraphrases the sentence by re-ordering or duplicating words into noun/predicative units. Since the approach identifies stratified layers of clauses, the scope of each negation cue is properly determined by using semantic tokens instead of syntactic punctuation marks. The authors offer as an example the sentence “Their menu was a good one that didn’t try to do too much” (p. 164). In this sentence, the negation cue “n’t” is under the sub-clause “try to do too much,” hierarchically related to “Their menu was a good one” through the conjunction “that.” It is worth mentioning that this solution can effectively model sub-clauses semantically dependent of the first part of the sentence because it does not relate the negation cue to the token “good,” as other solutions would do.

2012 SEM Conference: Shared Task on Scope and Focus of Negation Resolution

A separate research effort in modeling the scope and focus of negation deserves specific attention. In 2012, the first edition of the Joint Conference on Lexical and Computational Semantics (or *SEM 2012) had a shared task dedicated to resolving the scope and focus of negation (Morante & Blanco, 2012). One of the contributions of this conference was the introduction of annotated corpora that goes beyond a particular topic; until this conference, the only corpus annotated with negative expressions for machine processing was BioScope, from the biomedical field. These new corpora (dubbed “CD,” as an abbreviation of Conan Doyle), consisted of a set of non-subject specific texts (Arthur Conan Doyle story tales) annotated with three elements related to negation scope (namely negation cue, cue’s scope, and the negated event), as well as the negation focus. Here is an example from the corpus:

- [John had] **never** [said {as much} before]

Square brackets enclose scope of negation, bold letters indicate negation cue, underlined words show negated events, and curly brackets indicate focus of negation. These elements are explained further when discussing each particular task.

Scope of negation modeling task. SEM 2012 task #1 concentrated on negation scope and task #2 on negation focus. Morante and Blanco (2012) offer definitions for scope and focus taken from Pullum and Huddleston (2002), according to whom scope is “the part of the meaning that is negated” while negation in general is defined as reversing the true value of a sentence to its opposite. The following section discusses task #1 only, since negation scope is of interest to this study.

The shared tasks offered closed and open tracks for participants to choose from. Participants who chose the closed track could only use the tools provided by SEM 2012, i.e. the CD annotated corpora with its three components (training, development, and test corpora). Conversely, contributors choosing the open track could take advantage of any external tool at their disposal, such as external semantic parsers.

As mentioned above, the task of scope resolution breaks down into three sub-tasks, as follows:

- (i) finding negation cues or words that identify negation: these can be single words, such as “no,” or multiple words, as in the case of expressions such as “no longer;” they can also be spread across a sentence (as in the case of “neither… nor”); finally, negation cues can also appear as word-parts or affixes, like prefixes (i.e. “im-” in “impossible”) or suffixes (for example, “-less” in “careless”).
- (ii) resolving the scope of negation: this relates to identifying the token or tokens affected by the negation cue, as in the case of the tokens “John has... said as much before,” which is affected by the negation cue “never” in the example above; those tokens can appear close to each other or discontinuous, which adds a level of challenge to the task.
- (iii) determining the negated event or property: i.e. the particular episode, circumstance, quality, etc. within the scope of negation that constitutes the specific target of the negated cue (in the example mentioned above, the event negated is that John has said something). In this sense, the task concentrates on factual events only, ignoring non-factual events such as expressions of modality.

Evaluation measures. F1 scores (i.e., the harmonic mean between precision and recall) is widely used to evaluate the performance of classifiers (Liu, 2011). For the SEM 2012 task #1, the pre-defined set of outcomes to be measured are as follows: (i) cue match (named "cue"); (ii) scope-level with partial cue match ("scope ncm") where only some tokens are identified within the scope of negation; (iii) scope-level with strict cue match ("scope cm") that evaluates whether all in-scope tokens have been identified; (iv) identification of negated events ("negated"); (v) global negation measure ("global") that assesses whether cue, scope and event negation have all been correctly identified; (vi) scope tokens ("scope") that evaluates whether all tokens within the span of negation were found (a case that applies particularly in the instance of sentences with more than one scope of negation; in these cases, all tokens from all negation scopes for that sentence should be added up); and (vii) percentage of correct negation sentences ("cns") (Morante & Blanco, 2012)

Winning solutions, closed track. The best performing solution for cue detection was developed with The FBK (Fondazione Bruno Kessler) system. Chowdhury and Mahbub (2012, June) describe their solution as a two-step process: (i) data preparation by detecting affixes ("un-able," for example) and then proceeding to treat them as sub-tokens (i.e. *un*able*) that could potentially hold negation cues; and (ii) deploying a conditional random fields (CRF) classifier (using the MALLET toolkit[4]) to predict negation cues based on already identified sub-token information. Regarding scope detection, the UWashington system developed a ternary token classification task: part

[4] McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit. Retrieved from http://mallet.cs.umass.edu

of the scope, out of the scope, or neither. White (2012, June) describes employing a CRF classifier that has been trained using the gold standard data provided by SEM 2012. Such a classifier (which can be found in the MALLET toolkit as the sequence tagger in the SimpleTagger class) uses features according to three groups of data: regular token, cue token, and syntax tree.

Winning solution, open track. In this case, the best performing solution for both negation cue and scope (strict token match) detection were developed by the same team from the University of Oslo (UiO2). Lapponi, Velldal, Øvrelid, and Read (2012, June) explain that their solution also employed a CRF classifier (the Wapiti toolkit[5] developed by Lavergne, Cappe & Yvon, 2010, July) with features expanding to the closest distance of tokens to the left and right of the negation cue, token and PoS level forward and backwards bigrams and trigrams, and lexicalized PoS unigrams and bigrams. As mentioned above, the new contribution of this model is that it also takes into consideration semantic aspects of negation by representing syntactic elements as dependency graphs (essentially a directed tree) using the Stanford dependency converter which is part of the Stanford parser[6]. The goal of the directed tree representation was to model the syntactic dependencies between tokens and their corresponding negation cues. The researchers also modeled different layers of semantic information, such as morphological (affixal) cues, in and out of scope, and negation stops.

[5] http://wapiti.limsi.fr
[6] http://nlp.stanford.edu/software/nndep.shtml

Negation Detection in Social Media Text: SemEval Tasks

2014 SemEval, task 9: Sentiment analysis in Twitter. A second effective model for negation detection (developed as part of 2014 SemEval, the Conference on Semantic Evaluation Exercises, Task 9 on sentiment analysis of tweets), deserves special attention for its relevance to this research. The best performing solution in this task, developed by the NRC-Canada team (Zhu et al., 2014, August), was also the winning solution for the previous edition of SemEval. As the authors mention, focusing on a more discriminative approach to modeling negation was a major improvement between their 2013 and 2014 solutions. In particular, labeling different types of negation words bumped up the performance of the SVM classifier from 86.37 to 86.63 F-scores (Zhu et al., 2014, August). As previously discussed, the authors' 2013 solution involved pre-processing negated bigrams such as "not good" to turn them into unigrams like "good_NEG" to then feed the classifier as features. For the 2014 solution, the pre-processing task involved a more discriminative approach to negation. For instance, "acceptable" in the sentence "this is never acceptable," was processed as "acceptable_beNever," but in the sentence "this is not acceptable" the corresponding representation became "acceptable_beNot." This new pre-processing labeling schema is based on the principle that "never" and "not" are different qualifiers with distinctive impacts on negative sentiment and, consequently, should not be labeled identically (as their previous models would have done, labeling both sentences as "acceptable_NEG"). As we can see, the foundation of this discrimination in modeling negation involves treating different negation cues differently. In both successful approaches, Zhu et al.

employed a support vector machine (SVM) classifier that leveraged ngrams, parts of speech, and lexicon related features.

The approach taken by the SemEval NRC-Canada researcher team has been of special interest for the present research for two reasons: (i) the use of social media text as data to improve negation modeling; and (ii) the discriminative approach taken toward negation identification, which inspired further investigation of the approximate negators as one more distinctive type of negation.

Later SemEval tasks on sentiment analysis in Twitter and negation modeling. The 2015, 2016, and 2017 editions of the International Workshop on Semantic Evaluation (SemEval) expanded and diversified the types of tasks related to the sentiment analysis of tweets. The following section discusses the tasks[7] and results as relates to their contribution to negation modeling. It is worth mentioning that, in 2018, this International Workshop will no longer offer a task for the sentiment analysis of tweets but will offer instead a more granular set of tasks revolving around affect, emojis, and irony under the umbrella of “Affect and Creative Language in Tweets” (SemEval 2018, 2018).

Sem-Eval 2015, task 10. For this edition, four subtasks were offered: (i) subtask A: phrase-level polarity or contextual polarity disambiguation, aiming to determine the polarity (positive, neutral or negative) of a word or phrase within a tweet; (ii) subtask B: message-level polarity classification: this is the same as the previous subtask but at the

[7] Notice that in 2015, Twitter’s sentiment analysis became task 10 (previously numbered task 9), while in 2016 and 2017, the same task was numbered 4.

tweet level (instead of word or phrase); (iii) subtask C: topic-based message polarity classification – aiming to determine the polarity of the tweet message towards that topic; (iv) subtask D: detecting trends towards a topic with a five-point polarity scale of strongly positive, weakly positive, neutral, weakly negative, and strongly negative, and finally; (v) subtask E: degree of prior polarity – measuring the strength of association of Twitter terms with positive sentiment, with scores ranging from 1 or maximum association with positive sentiment (or least association with negative sentiment) to 0 or least relation to positive polarity (or maximum relation to negative one). In this later subtask, the term "prior polarity" refers to the polarity of a term as previously assigned in a lexicon created for the task; this lexicon offers, for the first time, polarity intensity instead of the discrete (positive, neutral, negative) labels typically offered by sentiment lexicons (SemEval-2015 Task 10, n.d.). As reported by Rosenthal et al. (2015, June), negation handling was among the most important strategies used by teams to improve performance. The following list discusses the negation solutions by lead winning teams along with their ranks in the subtasks:

- Negative emoticons signaling a tweet with negative polarity: tweets containing a negative emoticon such as ": (" are labeled with negative polarity (Severyn & Moschitti, 2015, June). This solution was used by the "unitn" team, which ranked 1st in subtask A and 2nd in subtask B (phrase and message level without topics);
- Negation contexts: bag-of-words heuristic solutions that reversed the polarity score of tokens following the cue, which are thus considered under the scope of negation. This solution was employed by team

> "KLUEless" (ranking 2nd in subtask A and C, 1st in subtask D, and 4th in subtask E), team "Lsislif" (3rd in subtask B and 2nd in subtask E), team "TwitterHawk" (1st in subtask C and 3rd in subtask D), and team "Webis" (1st in subtask B). Each implementation differed slightly, as follows: KLUEless added a "not_" prefix to each token after the negation cue for up to 4 tokens, thus reversing their polarity (Plotnikova et al., 2015, June); Webis also added a negation label but multiplied the polarity score of those tokens by -1; Lsislif added a negation suffix to all tokens from the negation cue up to the next punctuation mark (Hamdan, Bellot, & Bechet, 2015, June); and like Lsislif, TwitterHawk also added a suffix ("_neg"), not only until the next punctuation, but also until the next hashtag, at-mention, or URL (Boag, Potash, & Rumshisky, 2015).

Sem-Eval 2016, task 4. That year, subtasks A and B replicated those from 2015, but for subtasks C through E, classification has been replaced by quantification as the predominant machine learning strategy. This new approach was taken because major fields interested in sentiment analysis (such as political science or market research) needed to estimate the percentage of tweets that are positive or negative towards a topic, i.e. to calculate the distribution of sentiment classes regarding that topic in a set of unlabeled tweets (SemEval-2016 Task 4, n.d.). Classes are represented by a 2-point scale (for "positive" or "negative"), or by a 5-point one (for classes "very positive," "positive," "ok," "negative" and "very negative"). As a consequence of this machine learning modeling shift, these subtasks are now reformulated as follows: (i) subtask C: classification of individual tweets according to a 5-point scale; (ii) subtask D:

quantification of a set of tweets according to a 2-point scale; and (iii) subtask E: quantification of a set of tweets according to a 5-point scale. As reported by Nakov, Ritter, Rosenthal, Sebastiani, and Stoyanov (2016, June), the winning team modeled their solutions using deep learning, particularly convolutional and recurrent neural networks, as well as word embeddings. Although in Nakov et al.'s overall results discussion about the task there is no specific emphasis on negation modeling as a performance improvement strategy, some authors reporting their solutions in individual papers do mention negation modeling strategies as important for performance. The most used strategy is still negation contexts, i.e. the aforementioned traditional approach of locating a negation cue and reversing the polarity of subsequent tokens until a punctuation mark is found. Negation contexts were used by the following teams: "TwiSE," ranking 1st in subtask C (Balikas & Amini, 2016); team "PUT," ranking 4th in subtask C (Lango, Brzezinski, & Stefanowski, 2016); team "Tweester," ranking 1st in subtask B (Palogiannidi et al., 2016); team "QCRI," ranking 1st in subtask E (Da San Martino, Gao, & Sebastiani, 2016), and finally team "UNIMELB," ranking 3rd in subtask A (Xu, Liang, & Baldwin, 2016). A different negation modeling approach was designed by team "ECNU" (ranked 2nd for subtask C), whose solution consisted of 29 negation tokens gathered from the Internet and used later for an estimation of the frequency of negations in each tweet (Zhou, Zhang, & Lan, 2016).

Regarding the importance of negation for overall performance, team PUT reported that the results of feature importance estimated by the F-statistic ranked negated tokens in second place of importance, representing 12% of all selected features in negated n-grams. Team ECNU also concluded that negation and tweet-specific features are the only ones that help improve scores for all subtasks (Zhou et al., 2016). Finally, team TwiSE reported the need to improve their negation model due to the inconsistent punctuation shown in tweets.

***Sem-Eval 2017, task 4**.* 2017 presented the last traditionally organized sentiment analysis on Twitter. Subtasks were the same as in 2016, but added a new language (Arabic), new evaluation measures, and a script to obtain the user profile information for Twitter users that published tweets in the database (SemEval-2017 Task 4, n.d.). From the winning teams, “Tweester” (ranked 3rd in subtask B) and “TwiSe” (ranked 2nd in subtask E) employed negative contents, as teams in former tasks had (Kolovou et al., 2017; Balikas, G., 2017). Tweester also added polarity reversal signaled by other tokens (such as emoticons) beyond negation cues, tokens which (they speculated) may reverse the polarity of the first part of a tweet when positioned at the end of it, which may happen particularly in the case of irony, sarcasm, and humor (Kolovou et al., 2017). A particular case is offered by team “funsentiment” (ranked 3rd in subtask E), which intentionally moved away from negated contexts and developed a negation feature that counts the number of negation words in the tweet without using a lexicon. The negation words they detect are: “no,” “not,” “cannot,” “rarely,” “seldom,” “neither,” “hardly,” “nor,” “n’t” and “never” (Li et al., 2017).

Besides the three aforementioned teams, the top winning solution for subtask A, “DataStories” (Baziotis, Pelekis, & Doulkeridis, 2017) and the winning solution for all remaining subtasks, “BB_twtr” (reported by Cliche, 2017) focused on improving different aspects of deep learning modeling with no particular emphasis in negation features.

Although the newly released 2018 “Affect and Creative Language in Tweets” task[8] moves away from its traditional sentiment analysis formulation, which has always been the classification of individual tweets in discrete classes of positive, neutral or negative tweets, the fact that now two of those subtasks (subtask 3 or “V-reg,” using regression for predicting sentiment intensity; and subtask “V-oc,” based on ordinal classification for sentiment analysis) focus on the intensity valence of sentiment (expressed by any real number between 0 and 1 for task 3 and with a 7-point scale for task 4), reinforces even more the need to investigate nuances and diversity in the polarity effect of negation cues, which has been the main goal of the present research study.

Final Remarks

The reviewed literature on the linguistics of negation has helped us identify how approximate negators sometimes operationalize prototype negation (as “not” does). Additionally, the literature on computational linguistics showed us that valence shifters can explain and model negation for automatic detection by machines. Since approximate negators are a type of valence shifter, it is worth exploring their role in negation detection, given the fact that they can actually behave as full negation cues (with scope and negated event).

[8] http://alt.qcri.org/semeval2018/index.php?id=tasks

Chapter 3: Research Design

Research Question

As mentioned previously, this research study aims to investigate how negation operationalizes in Twitter utterances, focusing on the role of a particular sub-group of valence shifters, the approximate negators. Such a sub-group encompasses adverbs that have the particular quality of turning the semantic meaning of tokens under their scope into their opposite polarities, behaving like actual negation cues (such as “no” or “not”). Specifically, the study endeavors to answer the following research question:

> **Research question: How do approximate negators operationalize reversal shifting (i.e. prototype or full) negation in Twitter utterances (tweets) in the context of improving automatic detection of negation in natural language processing?**

Research Sub-questions

An effective methodology for finding an answer to a research question is to break it down into more manageable components (Creswell, 2014; Leedy & Ormrod, 2013). For this particular research study, a fundamental procedure consists of identifying when an approximate negator behaves as a full negator or reversal shifter, which allows it to then be labeled as a negation cue. As mentioned in the previous section, that decision can be made by identifying a series of recurrences that approximate negators follow when acting as negation cues (as described by Pullum & Huddleston, 2002). The first research sub-question was formulated to accomplish this task, as follows**:**

Research sub-question 1: In which ways do approximate negators reoccur when behaving as reversal shifters (prototype negators or negation cues) in tweets?

As described below, the answer came from human annotation performed on Twitter data looking for said recurrences and also recording observations related to their context or function. Additionally, after the approximate negator was identified as a negation cue, human annotators also recorded the scope or negation, i.e. the extent of its effect over other tokens in the tweet. This annotation task not only helped in confirming the negation nature of the token (by positively identifying its negated semantic components), but prepared the data to answer the second research sub-question, as explained below.

Another foundational component of the overall research question refers to the potential impact of this phenomenon on the automatic detection of negation, eventually improving sentiment detection by machines. This component focused on defining and implementing programming strategies to automatically detect approximate negators acting as reversal valence shifters (so they can be properly processed as negation cues), and then the scope of their negation effect. The second research sub-question addressed this task and was formulated in the following terms:

Research sub-question 2: How do we automatically detect approximate negators behaving as reversal valence shifters (or prototype negators) in tweets?

To answer this sub-question, a machine learning classifier was trained using features that are unique to the occurrence of said full valence shifters. The definition of those features came from manually annotating recurrences during content analysis in the process of answering the first research sub-question. Hence, the corpus of tweets annotated for sub-question 1 was used as the gold standard for this task.

Research Methodology

Twitter Data Related Issues

Data sampling limitations. A specific concern when handling Twitter data relates to the reliability of data samples extracted using publicly available Twitter APIs. According to Driscoll and Walker (2014), this concern relates to three recurring issues: (i) difficulty in reproducing data collection methods, mostly due to the ongoing fluctuation in Twitter's data harvesting processes; (ii) stability of technical affordances, as the platform periodically develops, discontinues, or re-names features, and; (iii) the lack of a shared vocabulary to effectively conceptualize tweets and their metadata. Added to these reliability issues, sampling accuracy can also be challenging, since tweets are not randomly distributed via those APIs. Rather, as Driscoll and Walker point out, "Twitter's internal software plays an editorial role in selecting and yielding tweets according to a set of heuristic algorithms that are not known by outside users" (Driscoll & Walker, 2014, p. 1749). In other words, different tweets can be gathered when replicating the same sampling mining model, which seriously affects reliability. Lack of documentation on how processing algorithms are developed by Twitter adds to the transparency issue surrounding this method of data mining. On top of this, and from a design methodology point of view, researchers must keep in mind that not all Internet users are also Twitter users. Moreover, the extent to which Twitter use can be considered representative of a particular overall population has also been questioned (Bruns & Stieglitz, 2014a). For instance, Twitter's penetration rates vary from country to country and across different demographic groups. Consequently, generalizations in the types of conclusions able to be reasonably drawn can also be problematic.

These elements should be carefully considered when handling tweets and drawing conclusions based on Twitter data. In the case of the present research, the mining methodology is thoroughly described, offering details of Twitter metadata harvested and technology affordances utilized, including a discussion of bot-generated tweets detected in corpora. For a language-based research venture such as this one, limitations in representativeness still apply. For instance, variation in penetration rates among English speaking groups (along with their idiosyncratic use of the language) undoubtedly affects the generalization of findings. It is also worth pointing out that differences among language groups according to locations are not currently reflected by Twitter. The "coordinates" parameter offered by the platform indicates geographical location, which may or may not directly relate to a particular language or dialect (as the most widely spoken language in a particular place may not be the language in which a particular Twitter user is most proficient).

Data Collection and Pre-processing

Since this study researched the behavior of particular approximate negators (initially "barely," "few," "hardly," "little," "rarely," "scarcely," and "seldom"), sampling procedures gathered tweets containing those words only. Although one option could have been to collect a random sample including tweets with and without approximate negators and to then separate sub-samples according to those words, for the sake of simplicity and focus, it was decided to gather data containing only those adverbs. Additionally, this study revolved around the reversal shifting behavior of these words and does not consider other general aspects of their behavior, such as how often they appear in data, so a wider sample would had been irrelevant to the research goals.

Data collection tool. Tweets were collected using “STACKS: Social Media Tracker, Analyzer, & Collector Toolkit” (Hemsley, Ceskavich, & Tanupabrungsun, 2014). STACKS is “an extensible social media research toolkit designed to collect, process, and store data from online social networks” developed by Syracuse University’s BITS (“Behavior, Information, Technology and Society”) Laboratory as part of their applications suite for conducting research on how people use information and communication technologies (http://bits.ischool.syr.edu/).

Following STACKS procedures[9], a Collector was installed to mine tweets using Twitter’s public Streaming API. The Collector ran on a server using the document-oriented database application MongoDB to store tweets. MongoDB is used to handle big sets of unstructured data, like that collected from social media sites for storage and querying.

Data collection procedure. The STACKS Collector was set up to create one list of tweets for each approximate negator word (as mentioned above, at the time of collection those words were: “barely,” “few,” “hardly,” “little,” “rarely,” “scarcely,” and

[9] https://github.com/bitslabsyr/stack/blob/master/INSTALL.md

“seldom”). Additionally, since the research focuses on the English language, only tweets holding the value “en” (for English) in the field “lan” (“language”) were collected. No other field parameters were established.

Data collection timeframe. The collection ran from 11/19/2015 at 1:55 PM through 12/28/2015 at 1:40 PM. In total, 18,368,195 tweets were collected and stored in a MongoDB database.

Sampling procedure. After mining approximate negator lists, tweets were extracted in two sets of 1,000 and 2,000 tweets per word. The sample procedure was not random but chronological, i.e. the first 1,000 tweets per word were selected. Those two sets of tweets underwent a data cleaning process (explained below) to arrive at a final set of 1,300 tweets per approximate negator.

Data cleaning procedure. The first cleaning procedure involved eliminating duplicate tweets, i.e. those tweets showing exactly the same value in the “id_str” field. Using this field for accurate tweet identification is advised in Twitter’s Documentation[10]. This procedure was performed during the mining process, in which those duplicate IDs were identified and their tweets discarded from the collection. The second data cleaning technique was applied using a Python program to eliminate retweeted text. The program searched for identical values in the “id” field within the “retweet_status” parameter. Finally, conversion into .csv files involved a third cleaning procedure focusing on removing tweets with exactly the same text content to avoid annotating duplicate data. A Python program was written taking advantage of regular expressions to eliminate exact matches of spaces and characters among different tweets, in order to

[10] https://dev.twitter.com/overview/api/twitter-ids-json-and-snowflake

keep just one version of the same text among different tweets. Thanks to this final procedure, text that was manually cut and pasted by Twitter users, thus generating duplicate content among tweets, was cleaned from the data.

As an example of the final result, after the three data cleaning procedures mentioned above were performed on the first 1,000 tweets per each word in the first phase of sampling (i.e. duplicate IDs, retweet, and duplicate text elimination), the final number of tweets per approximate negator stored in .csv files were: “barely,” 986; “hardly,” 980; “rarely,” 912; “scarcely,” 889; and “seldom,” 797.

Corpus development. As explained later on, the training phase discarded the approximate negators “few” and “little” as subjects of study; thus, the final corpus consisted of 1,300 tweets for each of the following tokens: “barely,” “hardly,” “rarely,” “scarcely,” and “seldom,” for a grand total of 6,500 tweets. Each tweet was assigned a “project ID” number consisting of a consecutive ordinal number preceded by the first two letters of the approximate negator’s word (i.e. “BA001,” “HA001,” “RA001,” “SC001,” and “SE001”) for further reference. To facilitate human annotation, tweets were converted into .csv files to be read in the Microsoft Office Excel spreadsheet application.

Annotation Methodology

The annotation task was performed in two phases: (i) tweet negation labeling and negation cue identification; and (ii) negation scope annotation. Two annotators performed the work: the graduate student researcher authoring this report and a Master’s student, under the supervision of a faculty advisor. Both annotators worked on tweets using an Excel spreadsheet with five columns: (i) tweet ID number, or the value from the “string_id” parameter in Twitter’s metadata; (ii) project ID number (such as “BA100” for tweet 100 in the “Barely” tweets sub-corpus), referring to the tweet location in the corpus; (iii) tweet text, with the content extracted from the tweet’s field “text;” (iv) negation label (“Partial,” “Full,” or “Ambiguous”), to be selected by the annotator; and (v) notes, for annotators to record observations or interpretations related to his or her decision. It is worth mentioning that the negation category column consisted of a pre-defined drop-down menu instead of a box for the annotator to type in the name of the category. This was a design choice because the literature suggests avoiding repetitive typing, which is prone to error (Pustejovsky & Stubbs, 2013). Consequently, annotators were only able to select one choice out of three, with a click, from the options given.

Training. The goal of the training phase was twofold: (i) developing the preliminary version of the Cue and Scope Guidelines, including annotators’ feedback; and (ii) making sure that both annotators interpreted those guidelines consistently throughout the process. Annotators worked in parallel on sets of 100 tweets per approximate negation word, for a total of 700 tweets per set. After each set of 700 was completed, disagreements in label assignment were discussed and clarified. The total training phase involved two iterations of 700 tweets each and was considered done when the inter-annotation agreement scores reached satisfactory values (as discussed below). Also, throughout the process, qualitative observations of negation phenomena were recorded for further exploratory analysis, identifying recurrences in the use of these approximate negation words and recording notes to assist in understanding.

After training, the final cue and scope annotation task started. Since two approximate negator cases were discarded as irrelevant to the study, which will be discussed shortly, the task was organized in 11 annotation iterations of 500 tweets each (100 per approximate negator), for a total of 5,500 annotated tweets. These tweets were added to the last 1,000 from the training set for a total of 1,300 tweets per approximate negator (6,500 tweets total in the corpus). The decision to add the second set from the training task was based on the optimal inter-annotator scores obtained (using Kappa scores, as explained below), with more than 0.70 agreement in all cases. In 4 of those iterations, a total of 2,000 tweets were annotated simultaneously by both annotators in parallel and their Kappa scores were processed to ensure maintenance of over 0.70, for reliability purposes.

Annotation-related emerging elements. Two new elements emerged during the annotation phase: (i) the words “few” and “little” showing insufficient occurrences as negators (which led to a reduction in the breadth of the study); and, (ii) identification of bot-generated tweets in corpora. The following section describes both elements and the research decisions made based upon them.

***Study breadth reduction: from 7 to 5 approximate negators*.** Although according to Pullum and Huddleston (2002), all seven words (five adverbs and two adjectives) are candidates to become full reversal shifters, the results of the preliminary study showed an uneven proportion of their reversal effect on Twitter utterances. In particular, the adjectives “few” and “little” showed a strong tendency to act as quantifiers or qualifiers rather than valence shifters. Moreover, during the annotation training phase not one of the 200 utterances containing “few” passed the negation polarity tests, while only one did so for the 200 utterances containing “little” (which was: “Little **do** they

know it was intentional," subject-auxiliary inversion test in bold). Furthermore, the later tweet sample included the expression "little reminder" twice, in which the use of "little" as an adjective actually intensifies the meaning of "reminder," since the utterer's intention is to politely emphasize that something needs to be reminded. The rest of the utterances presented "little" as also working as an adjective (like "hello my little princess" or "little brother asked me for a sharpie tattoo"). Finally, the determinative "few" was widely used as a quantifier with no negative shifting effect ("a few months," etc.) or as an adjective ("a few moments"). Due to the limited use of these words as negators, both of them were eliminated from the study, reducing the scope from 7 adverbs to 5.

Identification of bot-generated tweets in corpus. During the annotation process, a certain type of tweets (largely coming from the "scarcely" sub-corpus) proved unintelligible. Here is one example:

- "Not only was furniture confounded; there was scarcely anything left of body or mind by which one could (cont) https://t.co/v6MSOHJV99" (SC919)

In order to investigate this issue, efforts were made to identify whether such tweets came from bots (short for robots), particularly social bots. A social bot is "a computer algorithm that automatically produces content and interacts with humans on social media, trying to emulate and possibly alter their behavior" (Ferrara, Varol, Davis, Menczer, & Flammini, 2016). These social media bots fulfill a variety of useful functions, such as automatically aggregating content or responding to inquiries from customers (as when branded companies adopt them for customer care). However, with the increasing popularity of social media, malicious bots also take advantage of users by sending spam, malware, misinformation, and the like.

The literature reports several content-related traces of bot text, especially when dealing with malicious social bots. Particularly in social media, URL shorteners help disguise malicious websites disseminated in tweets (Chakraborty, Pal, Pramanik, & Chowdary, 2016). Furthermore, the mere presence of a URL may be evidence of a tweet sent by a bot (Yang, Harkreader, & Gu, 2011, September). Additionally, automatic bots rephrase the same content several times to increase the chance of getting attention. This technique of using so called “heterogeneous tweets” also helps bots bypass Twitter’s spam detection tools, which identify users who send a higher than standard number of retweets (Yang et al., 2011). These content-related bot footprints were found in the corpus, confirming that those unintelligible tweets could well have come from social bots.

The next step was to identify specifically which tweets came from social bots. Chu, Gianvecchio, Wang, and Jajodia (2012) report that the device from which a tweet has been sent helps to clarify whether its source is a bot. Twitter’s users can send their tweets via several devices: the Twitter website, mobile devices (Android or iOS), registered third party applications (such as TweetDeck), or via APIs. Since their goal is to automatically disseminate a high amount of information quickly, bots generally come from APIs; thus, finding API bot sources in tweets could be a straightforward way to identify these bot-generated tweets.

The Twitter documentation indicates that the field “source” contains information on the “utility used to post the Tweet, as an HTML-formatted string” (Twitter Developer

Documentation, 2017). Additionally, a post in the Twitter Developers Forum confirmed that this field has details about the name and URL of the application used to post a particular tweet (Twitter Developers Forum, 2014, February). Since the field “source” was selected to identify tweet bots, a Python program was developed to identify tweets in the corpus that had a URL in the “source” Twitter metadata field showing the string “bot.” This pragmatic decision was made because an inventory of bot sources at the moment of writing this document goes beyond the scope of this investigation. A comprehensive investigation of the bot phenomenon in social media is also beyond the scope of this document. The sources of bots continuously evolve and transform, and lists of identified bot sources continuously change as they become exposed.

After the data was passed through the programing solution, 195 out of the 6,500 tweets of the entire corpus (3%) had a bot URL address in the source Twitter parameter. The following table shows the breakdown of bot-URL tweets by approximate negator word:

Approximate negator	Bot-generated tweets	Percentage from total corpus	Percentage from AN’s sub-corpus	Percentage from total of bots
Barely	15	0.2%	1.1%	7.7%
Hardly	21	0.3%	1.6%	10.8%
Rarely	36	0.5%	2.7%	18.5%
Scarcely	88	1.3%	6.7%	45.1%
Seldom	35	0.5%	2.6%	17.8%
(Subtotal)	*(195)*	*(3%)*		*(100%)*

Table 2: Percentage of bot-URL tweets by approximate negator in each sub-corpus and total in corpora

As the table shows, “scarcely” holds the highest number of bot-originated tweets with 88 or 6.7% within its own sub-corpus (45% of all bot tweets), followed by “rarely” and “seldom” at a very similar rate, with 35 and 36 tweets or 2.6% or 2.7% of each sub-corpus respectively.

At the same time, these figures can be contrasted against trends in the way annotators categorized data. As it will be discussed in the “Cue annotation” section, annotators labeled tweets according to three categories: “Full” when the approximate negator acted as a prototype negation cue, “Partial” when this word was a regular valence shifter, and “Ambiguous” when the meaning of the tweet could not be decided. After the annotation process was finished, the breakdown of tweets labeled “Ambiguous” by sub-corpus, showed higher numbers in the “scarcely” set, where these bots are mostly present:

Sub-Corpus	# of tweets labeled as “Ambiguous”
Barely	8
Hardly	2
Rarely	5
Scarcely	443
Seldom	35

Table 3: Tweets labeled as “Ambiguous” by sub-corpus

However, although unintelligible text seems to be related to the presence of bot-originated tweets (or at least we see a trend regarding the sub-corpus with the highest number of unintelligible text also showing the highest percentage of bot activity), legible tweets also reflect the same trend. The following table offers the breakdown of bot-originated tweets by type of label assigned by human annotators:

Label Sub-corpus	Annotation Label			Subtotal
	Full	Partial	Ambiguous	
Barely	2	13	0	15
Hardly	10	11	0	21
Rarely	7	29	0	36

Scarcely	18	27	43	88
Seldom	11	23	1	35
Subtotal	48	103	44	195
% from bot group	**24%**	**52%**	**22%**	

Table 4: Number of bot-generated tweets by annotation label.

As the figures show, tweets with the label “Partial” comprise the largest rate with 52% of bot tweets labeled as such, followed by “Full” with 24%, while the label “Ambiguous” (highly skewed in the tweet set “scarcely”) shows the lowest rate with 22% of bot tweets being labeled this way. Although the representation of this type of bot (those with the world “bot” in the URL) in the corpus is low (only 3%), the performance of bots mirroring human language is highly effective, with a total of 77% (151 tweets) of the bot-originated tweets being understood by humans and labeled as either “Full” or “Partial.”

This exercise helped identify whether the approximate negator corpus contained tweets coming from bot sources. On social media platforms, the bot environment is growing in size and complexity. According to Chakraborty et al. (2016), 80% of spam is currently delivered via communication outlets using botnets. Bot activity and the nature of bots themselves is changing rapidly. For instance, a new, hybrid kind of bot called a “cyborg” involves a semi-automatic generation of a bot message, in which one part is automatic but there is also human activity. As mentioned above, bot messages can be benign tweets that disseminate legitimate information automatically, such as in the case of RSS feeds linked to Twitter accounts. Additionally, the existence of bots indicates automatic dissemination but it may not signal automatic generation of text; rather, it can indicate that human or hybrid created text is disseminated automatically (such as in the case of RSS feeds). Moreover, even in the case of automatically generated text (using tools such as Spinbot), such text attempts to follow human-generated syntax and semantic patterns.

Being that bots are a growing trend in social media activity, they are part of Twitter's data limitations described above. However, within the scope of this research work, as long as a tweet is intelligible, the fact that it may have come from a bot was not considered a factor for exclusion from the corpus. Moreover, the fact that an annotator could not distinguish between human- and machine- generated text only proves how highly effective natural language tools have become in mirroring human language. In that sense, if automatically-generated text mirrors the way humans understand and use approximate negators as full valence shifters, such text could be considered as usable data for the present analysis. Finally, the rate of bot-generated tweets detected by this exercise was only 3%, with gives a limited dimension to the importance of the issue.

Research Methodology for the Qualitative Methods Phase

Reliability and Validity

Ideally, the components of a research design are engineered with the goal of guaranteeing its reliability and validity (Krippendorff, 2013; Neuendorf, 2002). Reliability refers to making sure that a selected methodology will give the same results when applied multiple times by different people (Neuendorf, 2002). In the case of human annotation reliability, Cohen's Kappa scores were used during training and also annotation guidelines and corpus development seeking to obtain optimal scores in between 0.60 and 0.80.

The notion of validity relates to ensuring that the designed research method measures exactly the phenomenon the researcher claims to be measuring instead of something else (Krippendorff, 2013). The annotation guidelines required specific attention to validity for their role in the generation of the gold standard corpus. Since researchers already established procedures for the identification of similar phenomena (particularly Morante et al., 2011, May; and Councill et al., 2010, July), those documents were used as bases for elaboration of the present annotation guidelines. Additionally, findings gleaned from knowledge in linguistic theory, mainly by authors already mentioned in the previous literature review section (such as Pullum & Huddleston, 2002; Jespersen, 1917; Klima, 1964), were used to investigate new specific recurrences of negation. These findings, however, were put into test in the annotation process in which, whenever a candidate form of negation showed up, it was left for the human to decide whether a full negation was present or not. Indeed, although some of those negation forms reported by the literature were present in the corpus, they did not signal negation and were not labeled as negation cues. Conversely, new recurrent cases of negation have presented themselves in tweets, as discussed in Chapter 4. To ensure the validity of these new negation cases, they were passed by the so-called “it is not the case that” semantic negation test, stated by Morante et al.’s annotation guidelines. This test consists of replacing the negation cue with the expression “it is not the case that...”, adding the words presumably negated, and checking to see whether the new sentence conveys the same meaning as the original one. As an example, the tweet: “I literally have barely anytime to sleep at the moment” (BA247) was rephrased by the annotator as “It is not the case that I have anytime to sleep at the moment;” its meaning considered equally as negative as in the original phrasing (holding “barely” as negator), and finally annotated as “Full” or a negated content tweet. Regarding the validity of negation scope annotation, it was ensured by using best practices followed by researchers (Councill et al., 2010, July) for ungrammatical and noisy data (such as that found on social media), addressed in the following section.

An additional threat to validity was related to the language and cultural background of the graduate student researcher, a Hispanic annotator with a mother tongue of Spanish who is also proficient in English as a second language. Such a background could predispose this researcher to ethnocentric biases in elucidating people’s usage of language through the lens of her own Hispanic cultural worldview. In this particular case, annotation validity was ensured by performing ongoing consultations with English-born speakers to make sure she understood tweets in the way that a native English speaker would.

Annotation Guidelines

Two annotation guidelines were developed for this project: (i) “Cue Guidelines” helped decide whether or not an approximate negator acted as a negation cue, marking the word as such, labeling the corresponding tweet as “Full,” and recording observations on specific recurrences found; and (ii) “Scope Guidelines,” which served the purpose of deciding upon and marking the negation scope or tokens whose valence was reversed by the negation cue.

The “Cue Guidelines” were developed based on “CLIPs Annotation of Negation Cues and their Scope, Guidelines v1.0” (Morante et al., 2011, May), used by annotators to develop the standard English (Arthur Conan Doyle tales and Wall Street Journal articles) corpora for the “SEM 2012 Shared Task: Resolving the Scope and Focus of Negation.” These Cue Guidelines also incorporated elements from the linguistic theory of negation as discussed in the preceding literature review as well as notes from observations recorded during the preliminary study. Regarding scope annotation, although the “Scope Guidelines” also followed general lines described in Morante et al., due to the particular nature of Twitter data (often ungrammatical and containing poor syntax) these guidelines also emphasized Councill et al.’s approach of simplicity, which is to annotate the minimum number of tokens within the negation span, “covering only the portion of text being negated semantically” (Councill et al., p.53). Finally, feedback from annotators was incorporated during the training phase to reach the final version of those guidelines.

Evaluation Metrics

After both annotators finished each iteration, the annotation results were incorporated into an inter-annotator Excel spreadsheet designed for this research project, in which all labels were automatically processed into the following confusion matrix for Cohen’s Kappa score calculation:

Annotator (A or B)	**B-Partial shifting valence tweet**	**B- Full shifting valence tweet**	**B-Ambiguous**
A-Partial shifting valence tweet	***A / B Partial***	A Partial / B Full shifting	A Partial / B Ambiguous
A-Full shifting valence tweet	A Full / B Partial	***A / B Full shifting***	A Full / B Ambiguous
A-Ambiguous	A Ambiguous / B Partial	A Ambiguous / B Full	***A / B Ambiguous***

Table 5: Confusion matrix with types of labels for result processing

Cases of disagreement were isolated for further analysis by both annotators, who later discussed whether or not their disagreement came from different interpretations of the same information or discrepancies in the application of the Cue Guidelines. Following that discussion, annotators agreed on common interpretations of the Guidelines and/or clarified their narrative, and also improved or expanded their observations in the "Notes" field. Finally, the graduate student annotator adjudicated the corpus in cases of unresolved disagreements.

Inter-coder agreement coefficient. Cohen's Kappa coefficient is commonly used by researchers to evaluate inter-coder agreement (Pustejovsky & Stubbs, 2012). This coefficient can be used when the process involves two annotators, and it also helps in measuring their level of agreement while also considering random or accidental agreement (Neuendorf, 2002). The formula is expressed as follows:

$$\mathrm{k} = \frac{Pr_{(a)} - Pr_{(e)}}{1 - Pr_{(e)}}$$

Where $Pr_{(a)}$ represents the percentage of agreement observed in the confusion matrix, which is calculated by adding up the diagonal values (showing perfect agreement between annotators) and dividing for the total number of tweets processed:

$$Pr_{(a)} = \frac{AB\ PartialR + AB\ FullR + ABambiguous}{N\ (or\ number\ of\ annotated\ tweets)}$$

On the other hand, $Pr_{(e)}$ accounts for the predicted agreement in the hypothetical situation in which each one of those annotators selects a category randomly (such as when making a mistake). In order to compute the predicted agreement, it is required that the percentages for all categories assigned by both annotators are multiplied in several steps, as follows:

1. For partially shifted valence tweets:

$$Pr_{(e)_{partialR}} = \left(\frac{A\ Partial}{N}\right) * \left(\frac{B\ Partial}{N}\right)$$

2. For fully reversed valence tweets:

$$P(e)_{fullR} = \left(\frac{A\ Full}{N}\right) * \left(\frac{B\ Full}{N}\right)$$

3. For ambiguous tweets:

$$P(e)_{ambiguous} = \left(\frac{A\ Ambiguous}{N}\right) * \left(\frac{B\ Ambiguous}{N}\right)$$

4. Aggregating $Pr_{(e)}$values: $Pr_{(e)} = \ Pr_{(e)_{partialR}} + Pr_{(e)_{fullR}} + Pr_{(e)_{ambiguous}}$

Finally, all values are computed in the k formula, as follows:

k = $\left(Pr_{(a)} - Pr_{(e)}\right)$ / $\left(1 - Pr_{(e)}\right)$

Following current literature in natural language annotation for machine learning (Pustejovsky & Stubbs, 2012), the recommended agreement level is established between 0.60 and 0.80, which is considered to be “substantial” (Landis & Koch, 1977, reported by Pustejovsky & Stubbs). This level of agreement was reached in the second iteration of 500 tweets, after which the actual corpus annotation began.

Kappa score and reliability. The concurrent annotation technique applied during training also helped to ensure annotation reliability throughout the corpus development process, in which for every 200 tweets annotated for each approximate negator’s sub-corpus, 100 of them were annotated simultaneously by both annotators. This practice of continuing double annotation iteratively guards against annotator drift.

The following table shows the total annotation assignments per sub-corpus, using “barely” as example:

Sub Corpus “barely”		
Sub-set	**Tweet’s project ID**	**Annotator**
1	BA001-BA100	Annotators 1 and 2 (training)
2	BA101-BA200	Annotators 1 and 2 (training; first set in the final corpus)
3	BA201-BA300	Annotator 1
4	BA301-BA400	Annotator 2
5	BA401-BA500	Annotators 1 and 2
6	BA501-BA600	Annotator 1
7	BA601-BA700	Annotator 2
8	BA701-BA800	Annotators 1 and 2
9	BA801-BA900	Annotator 1
10	BA901-BA1000	Annotator 2
11	BA1001-BA1100	Annotators 1 and 2
12	BA1101-BA1200	Annotator 1
13	BA1201-BA1300	Annotator 2
14	BA1301-BA1400	Annotators 1 and 2

Table 6: Annotation assignment for sub-corpus “barely.” Other sub-corpora replicated these assignments.

Corpus Annotation Phase

Cue annotation. This task focused on identifying approximate negators that behaved as full valence shifters and labeling the tweets that contained them as “Full” (for full reversion valence) differentiating them also from those with a partial negation valence (tweets which were labeled as “Partial”) and finally recording as “Ambiguous” those tweets in which the meaning was equivocal. The approximate negators acting as prototype negators, along with specific supporting tokens that reinforced their negative import (as explained in Chapter 4), were annotated in bold to indicate their role as negation cues. Additionally, for each identified cue within the tweet, the annotators marked the tokens whose valence was affected in between square brackets (“[ ]”) as preliminary scope annotation. This preliminary scope annotation was necessary to help annotators discuss disagreements regarding negation interpretation, a process that involved analyzing which parts of the sentence were considered negated or not. However, since negation scope annotation is a task complex enough to require focused analytical work and also specific annotation guidelines, the final version of negation scope annotation was performed in the second annotation phase.

Scope annotation. Scope annotation was performed using the preliminary scope marked during the cue annotation phase. Annotators followed the same training methodology, consisting of two phases of 100-tweet sets per approximate negator word (500 tweets per phase, 1,000 tweets total as training material). Inter-annotation reliability was evaluated by calculating token overlap between annotators, i.e. which tokens were marked under scope by both annotators. Following Morante and Blanco (2012), two types of overlap were estimated: strict scope match, which occurs when both annotators select exactly the same tokens as part of the scope; and partial scope match, which occurs when they agree on only some tokens as part of the scope. The graduate student researcher developed a Python program to compute which tokens overlapped and, consequently, the number of tweets that showed partial or strict scope matches. Reliability was considered satisfactory whenever 80% of annotated tweets showed strict scope overlap in each 100-tweet set.

All concurrently annotated scope tweets (400 per approximate negator; 2,000 total)		
Approximate negator	Number of strict matches/total	Percentage Strict/total (accuracy)
Barely	59/61	96%
Hardly	133/142	93%
Rarely	67/74	90%
Scarcely	51/54	94%

Seldom	80/83	96%

Table 7: Scope annotation matching table summary

Total reliability scores for cue (Kappa) and scope (accuracy) annotation:

Set 4 (project IDs 301 through 400)

Approximate negator	Kappa	Number of strict matches/total	Accuracy
Barely	83.49	17/18	94%
Hardly	77.64	34/38	89%
Rarely	94.59	18/22	81%
Scarcely	87.43	14/15	93%
Seldom	75.57	21/22	95%

Table 8: Reliability scores for Set 4

Set 7 (project IDs 601 through 700)

Approximate negator	Kappa	Number of strict matches/total	Accuracy
Barely	96.37	15/16	93%
Hardly	85.14	34/34	100%
Rarely	96.68	18/18	100%
Scarcely	69.97	6/6	100%
Seldom	90.12	19/20	95%

Table 9: Reliability scores for Set 7

Set 10 (project IDs 901 through 1000)

Approximate negator	Kappa	Number of strict matches/total	Accuracy
Barely	93.89	17/17	100%
Hardly	82.39	34/38	89%
Rarely	87.65	15/18	83%
Scarcely	81.96	21/24	87%
Seldom	94.85	20/21	95%

Table 10: Reliability scores for Set 10

Set 13 (project IDs 1201 through 1300)

Approximate negator	Kappa	Number of strict matches/total	Accuracy
Barely	90.63	10/10	100%
Hardly	76.01	31/32	96%
Rarely	86.49	16/16	100%
Scarcely	81.67	8/9	88%
Seldom	93.15	20/20	100%

Table 11: Reliability scores for Set 13

The following constitutes an example of annotated tweet (note that the approximate negator “barely” along with the negation supporting word “anywhere” are marked as negation cues):

- In the house girl 💁‍♀️ [I **barely** go **anywhere**] (BA1372)

Fieldnotes and observations. Beyond assigning labels related to the extent of negation by approximate negators, the annotation procedure also recorded more overall observations about the negation phenomenon on Twitter in the form of fieldnotes. As stated by Tracy (2013), fieldnotes are “textual notes used as the basis for later research reports[; they] consciously and coherently narrate and interpret observations and actions in the field” (p. 128). Fieldnotes focused on particularities in the use of approximate negators, with a special focus on their use as prototypical negation cues. Additionally, codes as categories (Saldana, 2013) were employed in order to find overall themes across tweet sub-corpora. Codes such as “exaggeration” or “sarcasm” grouped tweets for which annotators understood the use of negation was associated with either overstating a fact or emotion or ridiculing a situation.

Research Design for Quantitative Methods (Machine Learning Experiments) Phase

The second research sub-question revolves around automatically detecting approximate negators acting as reversal shifters and their scope of influence. To accomplish this, annotated data can be used when running machine-learning experiments. This section details the research design decisions made and steps undertaken for this task.

Definition of Terms and Evaluation Metrics

We define the “gold standard” as the corpus of tweets labeled by humans according to the negation role that each approximate negation plays with respect to other tokens in the tweet. “Label” refers to the type of negation category assigned to each tweet during annotation. “Label” has three values: “Full,” representing full negation and assigned to those tweets that showed approximate negations performing a complete valence shifting effect, from positive to negative, over surrounding tokens; “Partial,” when the approximate negator stays a partial shifter; and “Ambiguous,” assigned whenever it was not possible to decide on either Partial or Full valence shifting. In the case of those tweets labeled as Full, we use the term “scope” to refer to the set of tokens whose valences are directly altered by the full negator.

Accuracy and types of errors*.* To evaluate the performance of our model, we follow common practices of using the evaluation measures of an information retrieval task, i.e. we interpret the classification task as identifying and labeling particular items that are part of a specific set. Following this procedure, the classifier needs to identify and select (i.e. retrieve) sub-sets of items, predict their labels, and then compare them to a target set (a human-annotated corpus or gold standard). Although tweets in our gold standard contain three sets of labels (Full, Partial, or Ambiguous), since the present experiments aim to answer the question “How do we automatically detect approximate negators behaving as reversal valence shifters (or prototype negators) in tweets?”, the focus of the classifier’s performance turns into predicting reversal valence shifters labeled as Full as distinguishable from the other two labels. Thus, for performance evaluation we grouped output results into two sets of labels: (i) Full on the one hand; and (ii) Ambiguous and Partial together on the other.

A basic evaluation measure is the percentage of items that the classifier labeled correctly from the total number of items processed in the corpus. This measure is called “accuracy” and, although easy to understand, it does not help in identifying the distinctive types of types of errors that occur when the classifier performs a task (Manning & Schutze, 1999); in our case, the types of mistakes made when assigning labels. However, identifying, analyzing, and weighting errors is an essential task for improving machine learning model performance (Jurafsky & Martin, 2009).

When the classifier fails to label items correctly, two types of errors occur: (i) items are wrongly rejected from the relevant set, like when a Full gets another label (type error I or false negative); or (ii) items are wrongly accepted within a set, such as Partial or Ambiguous tweets labeled as Full (type error II or false positive). Following Manning and Schutze (1999), we represent these errors within a set diagram:

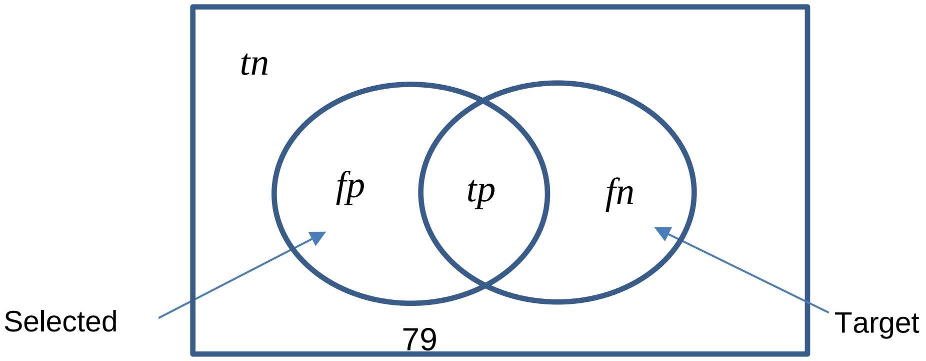


Figure 4: Types of errors in the classification task (Manning and Schutze,1999, p. 268)

The intersection represents the true positives (“tp”) or those Full tweets in the corpus labeled as Full by the classifier. To the right, we see the false negatives (“fn”) which are those tweets among the target set (i.e. Full tweets) that were mislabeled (explained by error type I). To the left, we see the false positives (“fp”) or tweets labeled as Full when they correspond to a different set (Partial or Ambiguous; error type II). Finally, true negative items (“tn”) correspond to true positives for other labels not under focus and, in consequence, out of the set diagram space (in this case, Partial tweets labeled as Partial, and Ambiguous tweets predicted as such).

As discussed, for the goal of this research, the focus is on those tweets labeled as Full, looking for true positives in this set and modeling features to reduce the number of false positives and negatives in this label class, as well as grouping true positives for other labels (e.g. Partial labeled as Partial and Ambiguous as Ambiguous) as true negatives for being out of the intended classification space. With that focus in mind, the calculation of false positives, false negatives, and true negatives for evaluating performance of the classifier will be as follows:

Predicted Actual	Ambiguous	Full	Partial	All
Ambiguous	*tn*	*fp*		
Full	*fn*	***tp***	*fn*	*fn*
Partial		*fp*	*tn*	
Average / Total		*fp*		*tn*

Table 12: Mapping between types of errors and label assignments for the classification task

Tables showing false positive, false negatives, and true negatives in Chapter 4 follow this approach.

Precision, recall and F1. As mentioned above, intuitive accuracy as an evaluation measure ignores the role played by different types of errors in a classifier's performance. To take those errors into account, we use instead "precision," "recall," and "F1," three measures that are widely used for information retrieval evaluation (Manning & Shutze, 1999).

Precision represents the ratio between Full tweets predicted correctly (true positives) and the total number of tweets predicted as Full, including type II error or false positive tweets (Partial or Ambiguous labeled as Full); in the following formula:

$$P = \frac{\mathrm{tp}}{\mathrm{tp+fp}}$$

Recall renders the proportion of Full tweets predicted correctly from the total number of tweets that should have been predicted as Full because they show that label in the gold standard. Thus, recall measures error type I or false negatives (Full tweets that have been labeled as Partial of Ambiguous); in the following formula:

$$R = \frac{\mathrm{tp}}{\mathrm{tp+fn}}$$

In terms of the confusion matrix results:

Gold Standard	**Classifier**	
	Predicted Full	**Not predicted Full**
Full	tp	fn
Partial or Ambiguous	fp	tn

RECALL

PRECISION

Table 13: Classifier’s evaluation measures with label assignments and types of errors

F1 is a combination of precision and recall with an added α factor for balancing biases that each one of these measures compute towards a particular type of result. Compared to a regular average, this added factor makes F1 a “harmonized measure.” Considering α = 5 as a popular choice for equal weighting of precision and recall, the F1 formula is:

$$\mathrm{F1} = \frac{2\mathrm{PR}}{(\mathrm{R}+\mathrm{P})}$$

F1 is widely used in the literature as a unified measure of machine learning effectiveness. However, once we obtain a score such as F1, how do we decide the importance of its value? What do we use to compare to or contrast against that score? Jurafsky and Martin (2009) introduce the concept of the “human ceiling” to answer this question. The human ceiling refers to scores obtained during the human annotation phase (inter-annotation agreement scores) for the same corpus, and it’s employed as a point of reference to evaluate the classifier’s performance. Considering that human annotators decide labels for the gold standard, comparing the performance scores of machines with those of human annotation when predicting labels can help researchers better understand how well automatic classification accomplished a task. The basic motivation for employing this method instead of comparing our results to an F1 score from another experimental setting is that labels, corpora, and types of tasks may differ in those settings.

All experiments used cross-validation as the model validation technique. Precision, recall, F-measures, and confusion matrices are reported with 5-fold cross-validation. Machine learning-related definitions of cross-validation and K-fold follow Jurafsky and Martin (2009). “Cross-validation” refers to the iterative process of selecting different sub-sets of tweets from the corpus to perform tests and training; “K-fold” refers to the number of times the iteration is performed; in our case, 5 times.

Machine Learning and Natural Language Processing Tools

Machine learning classifier. Previous research on automatic prediction of negation scope reported that the most effective classifiers for this task are support vector machines (SVM) and conditional random fields (CRF) (Morante & Blanco, 2012; Zhu et al., 2014, August). As discussed in the literature review, teams developing solutions for the scope prediction task in the SEM2012 competition employed both classifiers, with the winning team using CRF for sequential classification. However, more recent solutions showed that regular classifiers outperform sequential ones when handling noisy social media data, specifically tweets: both the 2013 and 2014 SemEval winning solutions for sentiment analysis of tweets have used SVM to improve negation modeling (Mohammad et al., 2013, June; Zhu et al. 2014, August). Additionally, this classifier has also shown optimal performance in online product reviews (Kiritchenko, Zhu, Cherry, & Mohammad, 2014, August; Cruz et al., 2016), and other born-digital text that is more casual than well-edited literary text. For the present research, the decision was to start with a regular SVM classifier (scikit-learn’s LinearSVC algorithm) to evaluate its performance before experimenting with sequential modeling as well. Since such performance was satisfactory (as reported in the experiment results chapter), no further model was developed; however, future work could also try conditional random fields (CRF) classification to further improve results. Finally, the SVM classifier was trained using lexical features as well as features particular to the phenomenon under research.

Natural language processing tools for Twitter data. Regarding text processing tools, both Stanford's CoreNLP[11] (Manning et al., 2014) and Carnegie Mellon's ARK TweetNLP[12] (Gimpel et al., 2011, June; Kong et al., 2014; Owoputi et al., 2013) performances were explored for tasks such as tokenizing, part-of-speech (POS) tagging, and dependency parsing because these are the only tools widely available for processing non-standard English text, such as Twitter utterances. The testing procedure consisted of several iterations of text processing using both toolkits on small data samples (between 4 and 20 tweets). In the case of Stanford's CoreNLP, caseless models were tested since they are suited for common formatting issues in social media and other colloquial, born-digital text (Stanford CoreNLP, n.d.). Regarding Carnegie Mellon's ARK TweetNLP suit, Twokenizer was employed for tokenizing and part-of-speech (POS) tagging, while TweeboParser (or Tweebo) served as a syntactic dependency parser. As confirmed through testing, Carnegie Mellon's ARK toolkit outperformed Stanford's CoreNLP due to better handling of Twitter affordances and common language usage practices. Common CoreNLP's labeling errors included the erratic labeling of at-mentions or hashtags as adjectives (JJ) or plural nouns (NNS) regardless of their function in a constituent; compound expressions (such as "idk" for "I

[11] https://stanfordnlp.github.io/CoreNLP/index.html
[12] http://www.cs.cmu.edu/~ark/TweetNLP/

don’t know”) wrongly categorized as verbs (VB); and mistaking separated constituents as one due to a tweet’s lack of punctuation, resulting in incorrectly training the classifier. Indeed, the literature reports that the ARK TweetNLP suit has been developed precisely to deal with the noisy, non-edited nature of Twitter language (Gimpel et al., 2011, June; Kong et al., 2014; Owoputi et al., 2013), while Stanford CoreNLP performs better with “well-edited English language,” as stated in their documentation (Stanford CoreNLP, n.d., ¶ Human languages supported). ARK’s Twokenizer, for example, deals with poor use of orthographic conventions, such as a lack of whitespace separation in between words (for instance, expressions such as “no:-d,yes” should be parsed out in four tokens; Owoputi et al., 2013), while their part-of-speech tagger uses a simplified and customized version of Penn Treebank and Wall Street Journal conventions, specifically suited for Twitter data (Gimpel, Schneider, & O’Connor, 2013). For instance, all verb forms (distinguished in Penn Treebank as VB such as VBD for a verb’s past tense, or VBG for a gerund) fall into the “V” label for simplification. Additionally, Twitter-specific tokens such as emoticons (symbols built with alphabet and punctuation tokens) and emojis (symbols rendered as small pictures) are labeled as “E,” and compounds (such as “lemme” for “let me”) receive an “L” tag. The 25 tags defined by ARK POS (from Gimpel et al.; 2013, p [1]) are as follows:

- Nominal tags:
    - N – common noun
    - O – pronoun (personal/WH; not possessive)
    - ˆ – proper noun
    - S – nominal + possessive
    - Z – proper noun + possessive
- Other open-class words tags:
    - V – verb incl. copula, auxiliaries
    - A – adjective
    - R – adverb
    - ! – interjection

- Other closed-class words tags:
    - D – determiner
    - P – pre- or postposition, or subordinating conjunction
    - & – coordinating conjunction
    - T – verb particle
    - X – existential there, predeterminers
- Twitter/online-specific tags:
    - # – hashtag (indicates topic/category for tweet)
    - @ – at-mention (indicates another user as a recipient of a tweet)
    - ~ – discourse marker, indications of continuation of a message across multiple tweets
    - U – URL or email address
    - E – emoticon
- Miscellaneous tags:
    - $ – numeral
    - , – punctuation
    - G – other abbreviations, foreign words, possessive endings, symbols, garbage
- Tags for other compounds:
    - L – nominal + verbal (e.g. i'm), verbal + nominal (let's, lemme)
    - M – proper noun + verbal
    - Y – X + verbal

Dependency parsing toolkit test. When having to choose a suitable tool for dependency parsing, a quick test was conducted on 20 tweets using both Stanford CoreNLP and ARK's Tweebo parsers as input. The goal was to choose the tool that included the highest number of tokens annotated as in-scope by humans as part of each negated cue's dependency graph. As a result of the test, Tweebo included in-scope tokens as expected for 17 tweets, while Stanford did it for 9. It is worth highlighting that not all tokens overlapped between human annotation and dependency parsing results; notwithstanding, Tweebo still showed better performance and was finally chosen for the experiments. As Kong et al. (2014) discuss, Tweebo handles particular aspects of Twitter utterances that affect dependency parsing, namely: (i) token selection that includes hashtags, URLs, emoticons, and other Twitter affordances;

(ii) multiword expressions that function as a single node in the dependency parse, such as idioms (“make sure”) or proper names with a high recurrence in Twitter data (“Justin Bieber”); (iii) the presence of multiple roots with their dependencies in each tweet, usually without clear punctuation marks to separate them from each other; (iv) a richer treatment of noun phrases’ internal structure by using direct dependency annotation instead of converting dependency structures from phrase-structure trees.

***Modeling a dependency path with Tweebo’s dependency graph information*.** An issue encountered when modeling dependency parsing for scope prediction was that ARK’s dependency parser does not include information on syntactic function between words, such as the direct object of a verb or its nominal subject. This syntactic information (called grammatical relations in the Stanford Dependencies Manual; De Marneffe & Manning, 2008) allows the creation of graphs by drawing a dependency relationship among words, i.e. in graph theory terminology, triples with nodes representing two words and an arc or edge that explains their relationship. According to dependency graph theory, the structure of a clause can be represented by binary and asymmetric relationships between pairs of words in that sentence. Simply stated, each word is related to another word as either the “head” or governor of that word or, conversely, its “dependent” or subordinated token (Nivre, 2010). Each head-node has a dependent-node via some particular type of asymmetric and syntactic edge relationship. For instance, a given verb token could be the head-node of a noun token as its dependent-node via a nominal subject edge or arc that explains what action (offered by the verb) that noun performs. Nivre (2010) offers the following example:

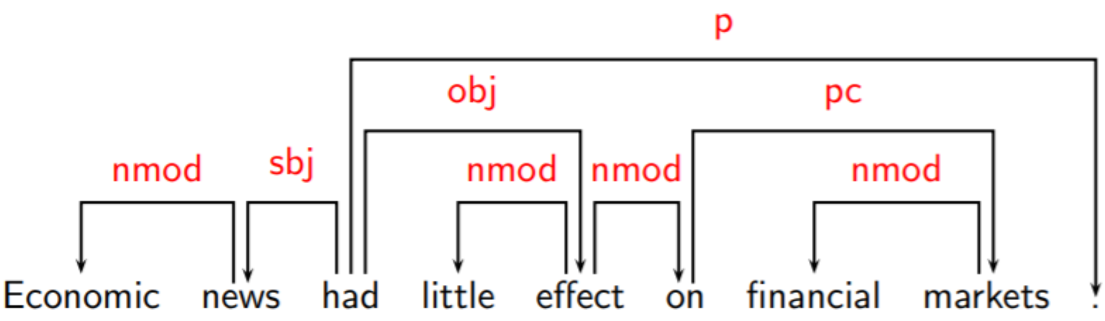


Figure 5: Example of dependency graph for a sentence (Nivre & Kubler, 2006, Introduction, slide 17)

In this sentence, the verb “had” functions as the head governing the noun “news” (its dependent) through a subject relation (“sbj”), while the same noun “news” becomes the head of adjective “Economic” thanks to a nominal modifier relationship (“nmod”); at the same time, the noun “effect” is the dependent of verb “had” in the role of direct object (“obj”) of that verb, and so on (Nivre, 2010).

As explained in the literature review, Lapponi et al. (2012, December) took advantage of these syntactic relationships to draw dependency graph paths based on graph information that effectively predicted the scope of a given negation cue. These authors offer the following example:

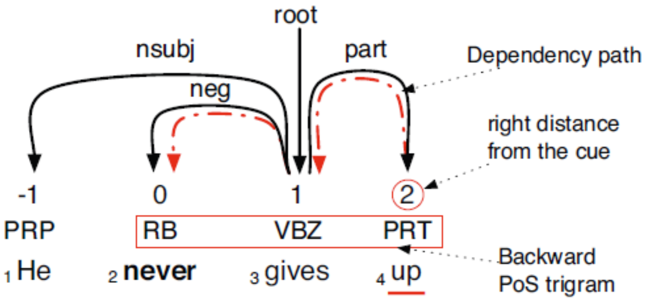


Figure 3 (repeated): Dependency paths for a negated constituent by Lapponi et al. (2012, December, p. 689)

Using the dependency syntactical relationship between head-nodes and dependency-nodes described in this graph's arcs (such as "nsubj" for nominal subject or "neg" for negation particle), dependency paths can be drawn from the negation cue, the adverb "never," to each token under its scope. Being that the phrasal verb particle "up" is under the scope of the adverb "never," the way to represent this in-scope dependency path is to go through the phrasal verb particle arc ("part") that relates that token to its head, the verb "gives," and use the negation modifier ("neg") relationship to finally reach the cue "never." Note that the red-dotted arcs, representing the in-scope path, are reversed from the original black arcs, with arrows pointing towards the head-token instead of the dependent. This reversion occurs because, in order to draw in-scope dependency paths, sometimes we need to walk the graph in the opposite direction from its asymmetric structure. In those cases, arcs with reversed arrows represent backward paths, i.e. from dependent to head. Lapponi et al. depict these two possible directions for the dependency path with upward ("↑") and downward ("↓") pointing arrows; thus, the final representation of this path becomes: ↑ "part" ↓ "neg."

Since Lapponi et al. proved that this way of modeling dependency paths as features helps improving the performance of a classifier for negation scope prediction, other state-of-the-art solutions have also taken advantage of it (such as Cruz et al., 2016), and so did this research project. However, as mentioned above, ARK's dependency parser (Tweebo) does not include rich standard syntactic information the way other dependency parsers for well-edited language (such as Stanford CoreNLP) can provide; instead, for each token, Tweebo provides its part-of-speech (POS) tag along with the index of its head-node token in the dependency graph. To compensate for this lack of information, a dependency graph solution was modeled, making the assumption that the POS tag of a dependent-node token (like “N” for noun) roughly represents the syntactic nature of the relationship between such a node and its head-node token (such as a verb token with the tag “V”). The “n” for “nominal” in “nsubj” “nominal subject” exemplifies this assumption. Additionally, the fact that this token-node is a dependent indicates the direction of the relationship, with an arrow coming from its head-node token. Following the nominal subject example, consider a node-token with ARK’s POS tag N being the dependent of a head with POS tag V; we could draw a triple that consists of an edge going from node V (head) to node N (dependent) and assume a nominal subject relation represented as V ↓ N.

In order to further illustrate how this solution helps developing features for scope prediction experiments, let us draw an in-scope dependency path for one of the tweets in the corpus using Tweebo’s output information and then represent that path as a feature for the classifier. We processed the following tweet (part of the “seldom” sub-corpus) as input for the parser:

- “@WndyCtyBsktball @michaelsobrien -actually they very seldom win city and state.” (SE165)

The following is the output dependency information offered by Tweebo. For each token, we receive its POS tag and the position (index number) of its head token in the sentence:

Token	POS	Token index	Index of token's head in dependency graph
@WndyCtyBsktball	@	0	-2
@michaelsobrien	@	1	-2
--	,	2	-2
actually	R	3	7
they	O	4	7
very	R	5	7
seldom	R	6	7
win	V	7	-1
city	N	8	9
and	&	9	7
state	N	10	9
.	,	11	-2

Table 14: Tweebo's output for tweet ID SE165

Note that the indexes "-2" represent a token out of the dependency graph, while "-1" signals the root of the graph, i.e. a node that has no further heads. With this parsing information, we can draw the following dependency graph:

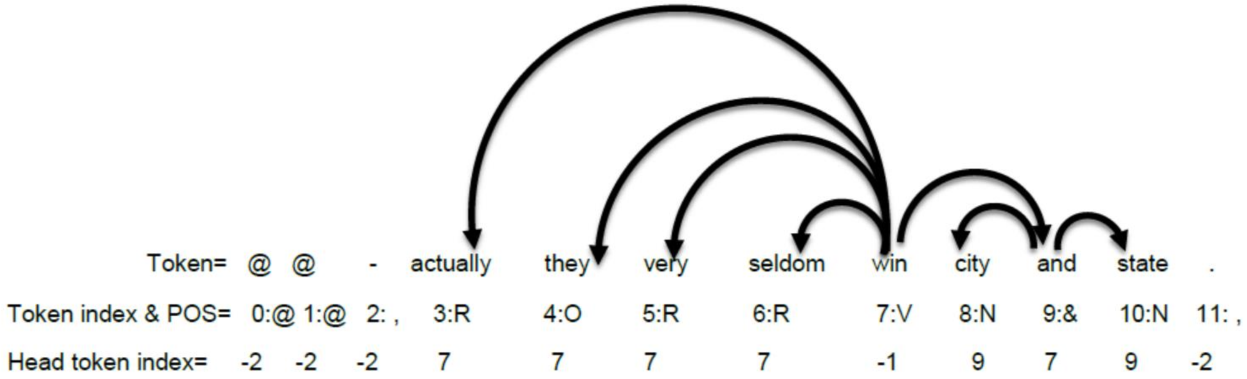
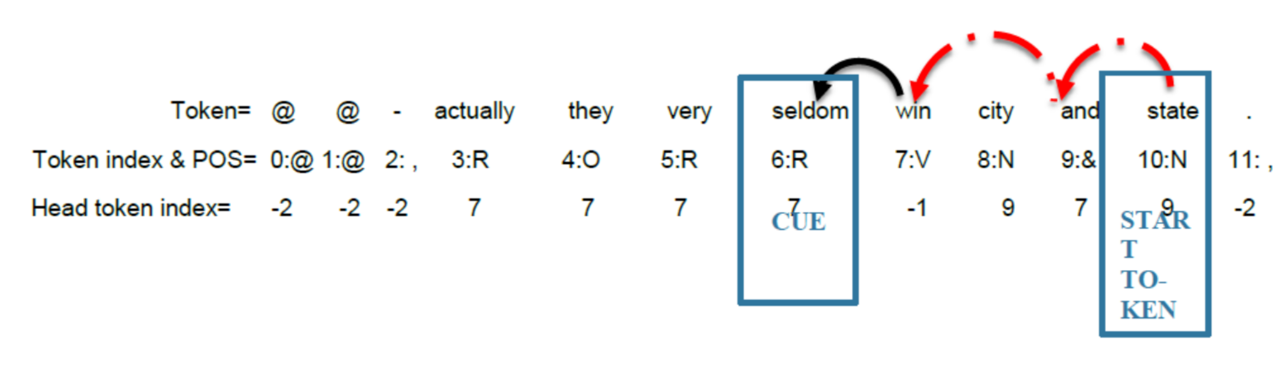

Token= @ @ - actually they very seldom win city and state .
Token index & POS= 0:@ 1:@ 2: , 3:R 4:O 5:R 6:R 7:V 8:N 9:& 10:N 11: ,
Head token index= -2 -2 -2 7 7 7 7 -1 9 7 9 -2

Figure 6: Dependency graph for tweet ID SE165

Note that each head token index value gives the index number for the head-node of a given dependent-node token, such as dependent adverb "actually," holding the verb “win” as its head-node at index 7. The token verb “win” is also the root of the clause (-1) while the at-mentions and punctuations are out of the graph (-2).

Using the information in this graph, we can now draw dependency paths from each token to the negation cue, which is “seldom” at index 6. Let us take the token noun “state” at index 10. In order to get backwards to the negation cue at index 6, the dependency path should go up the graph (using a reversed arrow) from “state” to its head, the conjunction “and” at index 7. Once there, we see that the conjunction “and” has the verb “win” for a head in position 7, so we reverse the direction of the arrow again to get there. Finally, since the negation cue “seldom” is a dependent of the verb “win,” we go down the regular arc path of this relationship to meet the cue at index 6. The final graph for this path is illustrated as follows:



Token=	@	@	-	actually	they	very	seldom	win	city	and	state	.
Token index & POS=	0:@	1:@	2: ,	3:R	4:O	5:R	6:R	7:V	8:N	9:&	10:N	11: ,
Head token index=	-2	-2	-2	7	7	7	7	-1	9	7	9	-2

Figure 7: Dependency path from token at index 10 to negation cue at index 7 for tweet ID SE165

Replacing syntactic information with POS tags for the destination token, the final dependency path from the token noun “state” to the cue “seldom” reads: “noun up to coordinating conjunction up to verb down to adverb.” Using Lapponi et al.’s nomenclature, it depicts as: N ↑ & ↑ V ↓ R. As an input feature for our classifier, this path was represented as: “Nu&uVdR,” where “u” represents an arc up the path, and “d” an arc down.

Following this solution, a Python script was developed that drew dependency paths from each token to the negation cue that shared the same root. Additionally, a sub-feature was added with information indicating whether the token was in the same phrase as the cue, defined as that token being in the phrase headed by the head of the cue. Both outputs were then offered to the model to make predictions.

Finally, scope information from the gold standard was delexicalized by replacing in-scope, cue, and out-of-scope tokens with the tags I, C, and O respectively, in variation of IOB chunking practices (Ramshaw & Marcus, 1995). In IOB chunking, tokens within a chunk are replaced by a specific tag indicating their position in that chunk, namely beginning (B), inside (I), or outside (O) of the chunk. The final set of tags depicts the delexicalized structure of that chunk. For the present research, the negation scope in a sentence was considered as a particular type of chunk with tokens inside or outside of it, also labeling the cue. The beginning (B) tag was not applied because, in the gold standard, each tweet with an approximate negator acting as a prototype cue presented one negation scope only—consequently, there was no need to indicate the beginning of several scopes corresponding to different negation cues. This could be explained due to the fact that tweets present short and simple sentences, with a reduced number of chunks on them.

Feature Engineering

We define features as properties that a model uses in order to make classification decisions related to words (McCracken, 2015). The following section describes the particular sets of features engineered for each specific classification task: (i) negation prediction, and; (ii) negation scope prediction.

Features for the negation (cue) prediction task. Specific features deemed instrumental for this research were identified and defined during the literature review and annotation phases, as will be discussed extensively in Chapter 4. Two types of features arising from the literature review are included: (i) features proven effective for modeling automatic detection of negation by previous research (particularly Lapponi et al, 2012, December; and Cruz et al., 2016); and (ii) features derived from patterns of negation discussed by the pragmatic linguistic literature and confirmed as relevant during the annotation phase. More precisely, rich annotation of observations confirmed the operationalization of specific types of negation that, although reported by the linguistic literature on standard English, needed to be tested in the colloquial English style present in social media. Finally, these rich observations also allowed the discovery of novel forms of negation operationalization.

The following is the list of features engineered for the classifier:

- Features from previous research on negation detection:
    - Lexical features: vocabulary size (baseline for experiments), bigram size, trigram size;
    - Twitter-specific features: number of hashtags, presence of at-mentions (binary value);
- Features from the literature review on negation phenomena:
    - Syntactic:
        - Reverse subject auxiliary (example: “scarcely ever do I see them running/walking for exercise”);
        - Reverse polarity tags (“..., isn’t it?”)

    - Word-related:
        - Approximate negator combined with tokens from the “any” word group;
- Features from annotation recorded observations:
    - Approximate negator combined with specific tokens (bigrams or trigrams):
        - “barely” ngrams: “barely_even,” “barely_ever,” “barely_ enough,” “barely_at_all;”
        - “hardly” ngrams: “hardly_even,” “hardly_ever;”
        - “rarely” ngrams: “rarely_if_ever,” “rarely_even,” “very_ rarely;”
        - “scarcely” ngrams: “scarcely_at_all,” “scarcely_if_ever,” “scarcely_enough;”
        - “seldom” ngrams: “very_seldom,” “seldom_if_ever,” “but_seldom.”

Most of these features, except for the reversed subject auxiliary and reversed polarity tags, were engineered using regular expressions. For reversed subject auxiliary and reversed polarity tags, part-of-speech tagging was employed as follows:

- Reversed polarity tag:
    - tags “V[13]” (verb) and “O” (pronoun) combined with the question mark (“?”) string: to capture tweets ending with expressions such as “would you?”
    - tags “L” (contraction of nominal and verbal) combined with the question mark (“?”) string to capture tweets ending with common casual contractions present in tweets such as “ist?”
- Reversed subject auxiliary: the following are the combination of part-of-speech tags identified in the corpus and used for the experiments. Each combination was represented with a variable named “pos_rsa” (which stands for “part-of-speech tag of reversed subject auxiliary”) with a consecutive ordinal number added:
    - Adverb, verb, pronoun, verb:
        - pos_rsa_1 = ['R,' 'V,' 'O,’ 'V']
        - example tweet: “@gabyhinsliff rarely have I been so glad to be /solely/ watching BBCQT through the medium of Twitter.” (RA1397)

[13] Description of ARK tags (Penn tags offered between brackets for reference): ‘A’: adjective (J*); ‘D’: determiner (WDT, DT, WP$, PRP$); ‘N’: common noun (NN, NNS); ‘V’: verb (including copula & auxiliaries (V*, MD); ‘R’: adverb (R*, WRB); ‘O’: pronoun (personal/WH; not possessive; PRP, WP); ‘^’: proper noun (NNP, NNPS).

- Adverb, verb, noun, verb, noun:
    - pos_rsa_2 = ['R,' 'V,' 'N,' 'V,' 'N']
    - example tweet: “And rarely did students take advantage of our services, and some claimed we were non existent. Naw, we holla'd at you, you never reached out” (RA1203)

- Adverb, verb, pronoun, verb, pronoun
    - pos_rsa_3 = ['R,' 'V,' 'O,' 'V,' 'O']
    - example tweet: “@MerrillLynch we barely make enuf to make ends meet how can they do that” (BA1266)

- Adverb, verb, pronoun, verb, noun:
    - pos_rsa_4 = ['R,' 'V,' 'O,' 'V,' 'N']
    - example tweet: “@KendallJenner every1 rarely do yall say shit directly I'm starting to catch on” (RA1128)

- Adverb, verb, noun, verb, determiner, noun:
    - pos_rsa_5 = ['R,' 'V,' 'N,' 'V,’ 'D,’ 'N']
    - example tweet: “Very seldom will someone enter your life and you won't have to questionÃ¢â,¬Â¦ https://t.co/NlNoa372T5” (SE746)

- Adverb, verb, noun, verb, determiner, adjective:
    - pos_rsa_6 = ['R,’ 'V,’ 'N,' 'V,' 'D', 'A']
    - example tweet: “What breaks my heart (and millions more) is the fact all war does is hurt,kill and maim the innocent. Very seldom does war kill the deserving” (SE1254)

- Adverb, verb, pronoun, verb, determiner, noun:
    - pos_rsa_7 = ['R,' 'V,' 'O,' 'V,' 'D,' 'N']
    - example tweet: “Very seldom do I fight the urge when I walk out of the gym to go over and get chicken house ðŸ˜, #BulkingSeason” (SE1030)

- Adverb, verb, pronoun, verb, noun, adjective:
    - pos_rsa_8 = ['R,' 'V,' 'O,"V,' 'N,' 'A']
    - example tweet: “Just means more PT for #Seldom, as in seldom does he do something positive. #kubball https://t.co/kK4Py3L3NK” (SE788)

- Adverb, verb, determiner, noun, verb, determiner, noun:
    - pos_rsa9 = ['R,' 'V,' 'D,' 'N,' 'V,' 'D,’ 'N']
    - example tweet: “Very seldom does a movie change your outlook on life for the better. Keep watching movies, they do so much for us.” (SE447)

- Adverb, verb, pronoun, verb, preposition or conjunction, determiner, noun:

    - pos_rsa10 = ['R,' 'V,' 'O,' 'V,' 'P,' 'D,' 'N']
    - example tweet: “Very seldom do you come across a company where you like everyone associated with it. Well, I have! I love... https://t.co/2IpPjMxDVi” (SE1155)
  - Adverb, verb, pronoun, verb, determiner, adjective, noun:
    - pos_rsa11 = ['R,' 'V,' 'O,' 'V,' 'D,' 'A,' 'N']
    - example tweet: “@colebrownpdx @carolynhanafee_ Yussssss! Very seldom do I give the perfect response!” (SE631)
  - Adverb, verb, proper noun, proper noun, verb, adverb, preposition or conjunction:
    - pos_rsa12 = ['R,' 'V,' '^,' '^,' 'V,' 'R,' 'P']
    - example tweet: “@TelegraphNews seldom do UK MPs vote quickly/decisively on contentious matters-too many howls of protest if we seem harsh/uncaring from left” (SE768)

Summary of POS reverse subject auxiliary tags:

- pos_rsa_1 = ['R,' 'V,' 'O,' 'V']
- pos_rsa_2 = ['R,' 'V,' 'N,' 'V,' 'N']
- pos_rsa_3 = ['R,' 'V,' 'O,' 'V,' 'O']
- pos_rsa_4 = ['R,' 'V,' 'O,' 'V,' 'N']
- pos_rsa_5 = ['R,' 'V,’ 'N,' 'V,' 'D,' 'N']
- pos_rsa_6 = ['R,' 'V,' 'N,' 'V,' 'D,’ 'A']
- pos_rsa_7 = ['R,' 'V,' 'O,' 'V,’ 'D,' 'N']
- pos_rsa_8 = ['R,' 'V,' 'O,' 'V,' 'N,' 'A']
- pos_rsa_9 = ['R,' 'V,' 'D,' 'N,’ 'V,' 'D,' 'N']
- pos_rsa10 = ['R,’ 'V,' 'O,' 'V,' 'P,’ 'D,' 'N']
- pos_rsa11 = ['R,’ 'V,' 'O,' 'V,' 'D,' 'A,' 'N']
- pos_rsa12 = ['R,' 'V,' '^,' ‘^,’ 'V,' 'R,' 'P']

Features for the scope prediction task. Feature engineering for scope prediction used lexical and dependency graph features reported in the literature (particularly Lapponi et al., 2012, December; and Cruz et al., 2016). Regarding dependency features, dependency paths were drawn using the aforementioned approach for ARK Tweebo’s dependency parser output. The final list of scope prediction features is the following:

Lexical features:

- Token name (baseline for experiments);
- Token POS;

- Bigrams: left and right;
- Trigrams: left and right;
- Bigrams' POS: left and right;
- Trigram POS: left and right;

Dependency graph related:

- Token-cue distance: number of tokens to the cue, left and right;
- POS of the first and second heads (for multi-constituent utterances): part-of-speech tag of the first and second order syntactic heads (i.e. heads of dependent constituents, such as in a nominal phrase) of a given token;
- Position of the token with respect to the cue: either cue being the ancestor of the token or token ancestor of the cue;
- Dependency path from token to negation cue: composed of two sub-features:
  - Token and cue sharing the same head in the head's dependency graph (values true or false);
  - Dependency graph path from token to cue: indicating POS tags for each token in the path along with the direction of the arc between nodes, such as up or down in the graph.

This chapter elucidated the qualitative and quantitative research design decisions for this inquiry. The following chapter will discuss the results of the content analysis and machine learning experiments that furnished the complementary elements required for providing an answer to the posed research question.

Chapter 4: Data Analysis and Results

Introduction

The following chapter reports two types of results:

(i) Qualitative content analysis: helped in answering the first research sub-question, stated as:

> Research sub-question 1: “In which ways do approximate negators reoccur when behaving as reversal shifters (prototype negators or negation cues) in tweets?”

An analysis of rich notes taken during the annotation phase along with elements from the preceding literature review shed light on ways in which approximate negators become prototype negation cues and also helped in the development of a list of recurrences of those negation cues. Later on, that list of recurrences supported feature engineering for informing the machine learning classifier when negation occurred in tweets.

(ii) Machine learning experiments: contributes quantitative results to answer the second research sub-question:

> Research sub-question 2: “How do we automatically detect approximate negators behaving as reversal valence shifters (or prototype negators) in tweets?”

Automatic detection of negation was modeled by engineering features to be used by a classification algorithm to predict: (i) whether or not a given tweet had full negation, and: (ii) which words (or tokens) within the tweet are being negated.

Qualitative Analysis: Content Analysis

Negation Recurrences Related to Negatively-oriented Polarity-sensitive Items

According to Pullum and Huddleston (2002), there are a series of tokens and phrases whose use is adequate only in the context of specific opposite polarities, i.e. either positive- or negative-valence constituents. They are called “polarity-sensitive items” and include words or groups of words such as particular phrases, idioms, etc. For instance, “already” is a positive polarity-sensitive item because it can be used in positive sentences such as “she knows him already.” When one wants to turn that sentence into a negative, “already” becomes unacceptable and should be replaced by a negative polarity-sensitive phrase such as “any longer,” resulting in the phrase “she doesn’t know him any longer” (Pullum & Huddleston, 2002; p. 822). Hence, “already” and “any longer” are part of distinguishing classes of polarity-sensitive items that are suitable for specific types of polarity only: “already” for positive polarity (hence called “positively-oriented polarity-sensitive items” or PPIs) and “any longer” for negative ones (taking the name “negatively-oriented polarity-sensitive items” or NPIs). In terms of Aristotelean logic, positively-oriented and negatively-oriented polarity-sensitive items belong to the contraries perimeter in the square of opposition discussed in the literature review (Chapter 2).

In the case of negative polarity, NPIs always show up in negative contexts, i.e. negative sentences or those holding prototype negators such as “not” or “no.” As the authors said, “all negators, whether expressing clausal or subclausal negation, sanction NPIs” (Pullum & Huddleston, 2002, p. 834). In other words, the presence of those NPIs reinforce the negative import of an existing prototype negation cue.

Although the class of NPIs is large, Pullum and Huddleston list the following items as the most important ones:

i.	The *any* class items: *any*$_{n}$[14]*, anybody*$_{n}$*, any longer, any more* (AmE *anymore*), *anyone*$_{n}$*, anything*$_{n}$*, anywhere*$_{n}$
ii.	Miscellaneous grammatical items (mostly functioning as adjuncts): *at all, either*$_{n}$*, ever*$_{n}$*, long*$_{n}$*, much, till/until, too*$_{n}$*, what(so)ever*$_{n}$*, yet*$_{n}$
iii.	The modal auxiliaries *dare* and *need*
iv.	A few lexical verbs: *bother* (+infinitival), *budge, faze*
v.	A large and probably open array of idioms, including: *can abide/bear/stand, can be bothered, could care less, cost a bean, do a (single) thing (about...), drink/touch a drop, eat a bite/thing, give a damn/fig, have a clue, have a penny (to one's name)* (BrE), *have a red cent,* (AmE), *hear/say a word/sound, hold a candle to, in ages, a donkey's year, lift a finger (to help), mind a bit, move a muscle, say a word, see a thing, see a (living) soul, so much as* (+verb), *take a (blind) bit of notice, would hurt a fly*

Table 15: List of negatively-oriented polarity-sensitive items (NPIs) by Pullum & Huddleston (2002, display [5], p. 823, italics from the original)

Pullum and Huddleston also establish a correspondence among positively-oriented polarity-sensitive items (PPIs), NPIs, and absolute negators. For example, the NPI "anybody" is a quantifier that negates the existence of somebody (due to the prefix "any"). As such, "anybody" corresponds to the absolute negators "no one" or "nobody" which, at times, both compare to "someone" as its related affirmative semantic (PPI) match.

The following is the full correspondence list offered by the authors:

PPIs	**NPIs**	**Prototype negators**
some	any	no
someone / somebody	anyone / anybody	no one / nobody
something	anything	nothing
somewhere / someplace	anywhere / anyplace	nowhere / no place
sometimes	ever	never
sometime, once	anytime, ever	never
somewhat	at all	
still	any more / any longer	no more / no longer
already	yet	

[14] The subindex "n" stands for "non-affirmative" meaning as differentiated from "f" or free choice meaning, which will be discussed in the next section.

so		neither / nor
too / as well	either	
	either	neither
	either... or	neither... nor

Table 16: Correspondence among positively-oriented polarity-sensitive items (PPIs), NPIs, and absolute negators (by Pullum & Huddleston, 2002, display [23], p. 831)

The case of "any" as a free-choice item. In their discussion about existential determinatives, Payne and Huddleston (2002) identify that "any" in a determiner function can take either the role of an NPI or what they call a "free choice" item, in which it conveys quantitative, polarity-neutral import. The following examples illustrate this use:

i Any computers with defective keyboards should be returned [plural]

ii Any policeman will be able to tell you [count singular]

iii Any remaining dirt will have to be removed [non-count]

(Payne & Huddleston, 2002, display [34], p. 382)

In this use, "any" indicates that there is a free choice because the predication property (i.e. being returned in the first example, telling the receiver in the second, and being removed in the third) can be applied to an arbitrary member of the group mentioned in the subject (i.e. computers, policemen, or remaining dirt). Hence, the use of "any" as a free-choice item conveys positive import in a sense that implies confirming the existence of something. The following examples show the free-choice use of this word:

i. We don't publish just any letters: we reject more than half of those submitted

i. Jan will read almost any computer magazines

(Payne & Huddleston, 2002, [display 35 and 36], p. 383)

Indeed, in their discussion of noun and pronoun phrases, Payne and Huddleston (2002) state that "any" has the same meaning as "some." The only difference is that "any" works only in non-affirmative contexts, which involve negative phrases but also interrogative ones; hence, the correspondence.

Notice that, in this function, "any" becomes an adjective modifying a noun or another adjective instead of an adverb. Such a role as a part of speech also indicates its function as a free-choice instead of a negatively-oriented polarity item. Interestingly enough, the literature indicates that approximate negators have a higher chance of fulfilling a stronger negation role when located at the beginning of the clause (Pullum & Huddleston, 2002, p. 820). In that sense, a clause opened by an approximate negator along with "any" could signal negation. However, no cases in the corpus demonstrated this scenario.

First type of recurrences of negation cues: Use of negatively-oriented polarity-sensitive items to reinforce the negative import of approximate negators in tweets. In the particular case of tweets, NPIs were found to contribute to increasing the negative import of approximate negators. As previously discussed, in prototype negative cases, the clause would show the prototype negation cue and the NPIs. However, some Twitter users decided to replace a prototypical negation cue with an approximate negator which, combined with specific NPIs, function as prototype negators by turning the polarity valence of nearby tokens to their opposite.

Here are some correspondences found in the data. As we can see, "any" and "ever" are the most common polarity items reinforcing the negative import of approximate negators:

PPI	NPI	**Reoccurring combination of NPI and AN**	Total number of tweets (Support)	Corresponding prototype negator
some / someone / somebody / something / somewhere / someplace	anything / anywhere / anyplace	barely any (123)[15], hardly any (247), rarely any (116), scarcely any (55), seldom any (56)	597	no / no one / nobody / nothing / nowhere / no place
sometimes / sometime, once	ever / anytime, ever	barely ever (14), hardly ever (151), rarely (if) ever (75), scarcely...(if)...ever (31), seldom (if) ever (56)	327	never
somewhat	at all	barely at all (6), scarcely at all (4)	10	

Table 17: Most common combinations of NPIs and approximate negators signaling full negation in corpora

As explained in the review of literature in Chapter 2, approximate negators are highly contextual in the sense that their role is defined by other words under their scope. “Hardly,” for instance, means “with hardness, i.e. with difficulty,” while “scarcely” acts as a “restricted negative” (Jespersen, 1917, p. 39-42). Following the same phenomenon, the combination of these words with NPIs such as “any” or “ever” turns the polarity of neighboring words into their opposite, thus signaling full negation. In natural language processing terms, the valence shifter tokens “hardly” and “scarcely” (approximate negators in linguistic terms) combined with the negative polarity tokens “any” and “ever” turns those shifters into negation cues that fully reverse the polarity of other tokens under their scope into their opposite valence.

It is worth mentioning that, although some of the NPIs mentioned by Pullum and Huddleston were present in tweets in combination with approximate negators, their

[15] Numbers in between parenthesis show the total number of reoccurrences in each sub-corpus

usage was rare and they were not labeled as recurrences by annotators. Two examples are “either” and “yet;” some examples follow:

- “You're young, you've barely got a chance to experience life yet.” (BA305)
- “@Latrobefanatic I rarely miss them either.” (RA810)

There were 11 cases of approximate negators combined with “either” to convey full negation; with “yet,” there were just 2.

***Negation emphasis*.** In spite of not being part of Pullum and Huddleston’s PPI-NPI-Prototype Negation table, a few cases of recurrences found during annotation seem to follow a behavior similar to that of negatively-oriented polarity-sensitive items as well. As Van der Auwera (2011) explains, linguistic researchers agree that languages find multiple ways to emphasize negation, one of them being through adding negative polarity particles to negative cues. This tendency originates from what researchers call Jespersen’s cycle. This cycle states that, in the history of languages, negation adverbs underwent a process in which they were first weakened in order to subordinate them to the main notion of the sentence; however, the utterer was compelled to then strengthen the negation effect by adding some other support particle, so s/he could make sure that the hearer received the negative message. Examples of final forms of Jespersen’s cycle are the use of auxiliary verbs in English negation (such as “do” in “I do not know”) and particles like “pas” in French negation (“Je ne sais pas”) (Jespersen, 1917; Horn & Kato, 2000). In the context of the present study, three specific words seemed to fulfill this negation emphasis role: “enough,” “even,” and “very.” The presence of these additional particles emphasizing negation could explain why the following recurrences were identified as negation by human annotators:

- “very” (216 recurrent occurrences): “very rarely,” “very seldom”
- “even” (130 recurrent occurrences): “barely even,” “hardly even,” “rarely even;”
- “enough” (26 recurrent occurrences): “barely enough,” “scarcely enough;”

The use of the conjunction “but” as concessive contrasting in “but seldom.” Jespersen (1917) discusses that the conjunction “but,” used as a negative relative pronoun, is etymologically related to “without” and, hence, conveys similar negative import. Horn (1989) expands this discussion indicating that the use of “but” in contrastive environments indicates contradiction, as in the following example from the Gospels of the New Testament:

- “Do not store up your riches on earth, where moths and rust destroy them… but store up your riches in heaven…” (Horn, 1989, display 78, p. 402).

In this example, the use of “but” establishes an opposition between two options: storing “riches on earth” or “in heaven.” In the case of the Twitter gold standard, cases of contrast using “but” are recurrent in the “seldom” sub-corpus (with a total of 109 reoccurrences). Here some examples:

- "True loves are often sown, but seldom grow on ground. #ALDUBSumptuousLunch" (SE1068)
- ”-tyrion: “People often claim to hunger for truth, but seldom like the taste when it’s served up.” -... https://t.co/XITTvx9N3m” (SE1227)
- “@jhazan I try my best, but I'm seldom that successful!” (SE1341)

In the first example, the utterer expresses the contradiction of “sown loves” that are not meant to grow; in the second, the message is that those people who seem to yearn for something actually do not enjoy it when received; finally, in the third case, the Twitter user expresses that the level of effort s/he has invested in doing something does not relate to the final outcome s/he obtains from that effort.

Negation Cases Related to Expressions of Negation

Pullum and Huddleston (2002) explain that negative polarity in a sentence can be indicated by one of the following types of clauses (shown with examples):

- Clause continuation with “not even:” “He didn’t read it, not even the abstract;”
- Connective adjuncts: “He didn’t read it; neither/nor did I;”
- Reversed polarity tags: “Ed didn’t read it, did he?;” and,
- Subject-auxiliary inversion with prenuclear constituents: “Not once did Ed read it.”

For the case of connective adjuncts, “either” is also mentioned in the literature as signaling negation by authors such as Klima (1964) and De Swart (2010).

Tweets labeled as negative in the corpus often followed one of those expressions of negation, although they became recurrences in only two cases: reversed polarity tags and subject-auxiliary inversion with prenuclear constituents. There were no cases of clause continuation with “not even” associated with an approximate negation in the corpus. In relation to connective adjuncts, “either” did show up along with an approximate negator signaling negation, but not often enough to become a recurrence (4 cases only). These four rare cases are shared for purposes of illustration:

- “An emotional 'Night with the Stars' @OasisAcademySP with my Son.Where has his last 5 years gone? No longer a child, hardly an adult either!” (HA245)
- “@ECigologist @LBC nice one Ian. I could hardly believe his words either.” (HA1265)
- “@Latrobefanatic I rarely miss them either.” (RA810)
- “@PaulbernalUK On other hand, Corbyn's position on how to stop ISIL - someone else do something - is scarcely convincing either.” (SC730)

It is worth highlighting that, in the third example, the utterer borrows the format of the everyday negative expression “I can’t believe…” and replaces the prototype negation cue and negated auxiliary verb combination “can’t” with the approximate negator “hardly.” This style of substitution has also been recurrently seen throughout the corpus.

Despite this, throughout the corpus “either” was used for reinforcing the negation import already contributed by another word and already combined with the approximate negator, such as “any:”

- “@aboycalledluke no it was quite nice actually, hardly any people either!” (HA1273)

The following sections will discuss reversed polarity tags and subject-auxiliary inversion with prenuclear constituents as taking the shape of negation.

Second type of recurrences of negation cues: use of reversed polarity tag. The pragmatic linguistic literature on negation also discusses the use of what is called reversed polarity tags, as in the example: “Ed didn’t read it, did he?” (Pullum & Huddleston, 2002, display [1], p. 786). In these cases, the main sentence (i.e. “Ed didn’t read it”) is called the anchor clause and the polarity tag (i.e. “did he?,” always attached to an anchor clause), can be positive or negative, constant or reversed. A constant polarity tag maintains the polarity of its anchor, while a reversed tag contradicts that polarity. The use of constant polarity tags is less frequent and more acceptable in positive clauses, i.e. along with positive anchors (Huddleston, 2002). Indeed, the use of reverse polarity tags with negative anchors is the most accepted use of polarity tags, to the point that these authors include it as one of the standard tests for negation (Pullum & Huddleston, 2002). Regarding its usage, an utterer may add a reversed polarity tag to an anchor clause with the purpose of seeking confirmation from the hearer. In that role, these tags are what Bolinger calls "echo contradictories" (1972, quoted by Horn, 1989) because the utterer seeks to echo his opinion with the hearer by stating a contradictory phrase. Metalinguistically (i.e. when considering contextual factors in which utterances are created), this contradictory functionality can signal an ironic use of language, as in the following example:

> "a. You aren't {slightly / just the least bit} tipsy, are you? (= You are…, aren't you)" (Horn, 1989, display 76, p. 402)

This type of negation also showed recurrently in the tweet corpus. As examples, we can see in the following tweets how the positive polarity tags "is it?" and "can we?" strengthen the negative valence of the approximate negators "hardly" and "scarcely" in their anchor sentences:

- "@HoeTurner @SuperIncognegro @ULT_VARAS @_buckyeahh Dude, it's IN the text of the Quran. That's hardly up for debate is it? :I " (HA447)
- "@gertsen11 It would've been considered escalation then. Now we can scarcely object, can we?" (SC527)

Third type of recurrences of negation cues: use of reversed subject-auxiliary syntax. As Haegeman (1995) and Pullum and Huddleston (2002) explain, negative words trigger subject-auxiliary inversion when they are positioned before the sentence's nucleus, as in the case mentioned above: "Not once did Ed read it" (Pullum & Huddleston, 2002, display [1], p. 786). In this example, the negator "not" is positioned before the nucleus of the sentence, which is "Ed." However, since negation relates to the main verb (the act of reading) and at the same time the English language tends to keep negation particles next to the verb they negate, the utterer is forced to use an auxiliary to make sure the hearer understands that the action is negated and not something else (such as the subject “Ed” or the object “it”).

Reversed subject-auxiliary syntax signaling an approximate negator acting as a prototype negation cue also became a recurrent phenomenon in the gold standard, particularly in the case of “hardly,” “rarely,” “scarcely,” and “seldom” (“barely” did not show occurrences). Here are some examples:

- “I always have the rudest things to say hahaha, hardly ever is it anything nice.” (HA740)
- “@holy_capp @oacapp very rarely do I see Owen ever driving” (RA383)
- “"He cannot die badly who lives well; and scarcely shall he die well who lives badly." -Augustine” (SC1135)
- “When you possess great treasures within you, and try to tell others of them, seldom are you believed” (SE559)

In the first tweet, for example, the utterer declares in the opening clause that s/he shares impolite statements all the time; it is assumed, thus, that the second clause really means that s/he does not express pleasant ones. This full negation assumption is operationalized by the reversed subject-auxiliary expression “hardly ever is it,” along with the NPIs “ever” and “anything.” Another NPI, “ever” supports the negative import of the reversed subject-auxiliary structure “rarely do I see” in the second sentence. Indeed, the presence of NPIs can only emphasize the negative operationalization of these reversed-subject auxiliary structures. In the case of the third tweet, there is a parallel structure between the two parts of the clause for which the negative import of the first part (stating that someone who lives well will not die poorly) is carried out by the second part when using the approximate negator “scarcely” which, along with a reversed subject-auxiliary structure “shall he die,” acts as prototype negation to express that equally, the one who lives a bad life will not die well. Finally, in the fourth example offered, the utterer expresses that people who share important messages are often not believed, probably for the high magnitude of those things (i.e. “treasures”), which makes them unbelievable. That un-believable nature of said things signals the negative import of “seldom” accompanying “believe” in the constituent that follows. This preceding negative clause, along with the reversed subject-auxiliary “seldom are you believed,” make the approximate negator behave as a prototype negation cue.

Closing Remarks: Answering the First Research Sub-question

This qualitative analysis section elaborated an answer to the first research question: “In which ways do approximate negators reoccur when behaving as reversal shifters (prototype negators or negation cues) in tweets?” The answer to this question involves the following reoccurrences of approximate negators when behaving as reversing valence shifters:

- Reoccurrences from the literature review and confirmed by human annotation:
    - Semantic:
        - Approximate negator combined with tokens from the “any” word group as NPI (negatively-oriented polarity-sensitive items, see discussion in previous sections);
    - Syntactic (derived from standard expressions of negation):
        - Reversed subject auxiliary (example: “scarcely ever do I see them running/walking for exercise”);
        - Reverse polarity tags (“..., isn’t it?”)

- Reoccurrences from annotation recorded observations:
    - Approximate negator combined with specific NPIs:
        - NPIs combined with “barely:” “barely even,” “barely ever,” “barely...enough,” “barely at all;”
        - NPIs combined with “hardly:” “hardly even,” “hardly ever;”
        - NPIs combined with “rarely:” “rarely (if) ever,” “rarely even,” “very rarely;”
        - NPIs combined with “scarcely:” “scarcely at all,” “scarcely (if) ever,” “scarcely enough;”

- NPIs combined with “seldom: “very seldom,” “seldom (if) ever,” “but seldom”

The next section describes the outcome of machine learning experiments when engineering these recurrences into features that, along with other features borrowed from the natural language processing literature, trained a classification algorithm to predict when a tweet having an approximate negator is actually a negated tweet, along with its negation scope on the tweet, thus answering the second research sub-question.

Quantitative Analysis: Machine Learning Results

This chapter discusses machine learning experiments and their outcomes; this discussion aims to answer the second research sub-question:

> Research sub-question 2: How do we automatically detect approximate negators that behave as reversal valence shifters (or prototype negators) in tweets?

The experiments tackle two aspects of automatic detection: (i) negation detection, defined as predicting whether a tweet is negative due to the presence of an approximate negator acting as full reversal shifter or negation cue; (ii) the scope of negation or the range of influence of the full negators within tokens in the tweet. Experiments related to scope negation will be discussed separately in the second part of this chapter.

Experiment Results for Automatic Detection of Negation

This automatic detection task was defined as a classification task for which each tweet containing an approximate negator had to be labeled as either Full, Partial, or Ambiguous according to a series of features (listed below). 1,300 tweets were classified for each of the 5 approximate negators for a total of 6,500 processed tweets. 5-fold cross validation was employed for precision and recall.

Features. The set of features used for this task included two types of standard features borrowed from the literature: (i) lexical features, such as unigrams, bigrams and POS tags; (ii) Twitter-specific features, such as the number of hashtags and at-mentions. Additionally, the negation recurrences listed in the qualitative results section were engineered from scratch using token and part-of-speech information. It is worth highlighting that, as described in the qualitative section, these negation-specific features indicated full negation in most cases, but not categorically, i.e. they were most often—but not always—unequivocally considered to be signaling negation.

This list of features is offered in this section for reference; a thorough discussion of these features can be found in Chapter 3 in the section titled “Features for the negation (cue) prediction task.”

- Features from previous research on negation detection:
    - Lexical features: vocabulary size, bigram size, trigram size;
    - Twitter-specific features: number of hashtags, presence of at mentions (binary value);
- Features from the literature review on negation phenomena:
    - Syntactic:
        - Reversed polarity tag: with two combinations of POS and tokens in the last three index positions of the tweet:
            - ‘V,’ ‘O,’ and the question mark character (‘?’);
            - ‘L,’ ‘O,’ and the question mark character (‘?’);

        - Reverse subject auxiliary: 12 part-of-speech tag combinations[16]:
            - Combination 1 (“pos_rsa_1”): 'R,' 'V,' 'O,' 'V'
            - Combination 2 (“pos_rsa_2”): 'R,' 'V,' 'N,' 'V,' 'N'
            - Combination 3 (“pos_rsa_3”): 'R,' 'V,' 'O,' 'V,' 'O'
            - Combination 4 (“pos_rsa_4”): 'R,' 'V,' 'O,' 'V,' 'N'
            - Combination 5 (“pos_rsa_5”): 'R,' 'V,’ 'N,' 'V,' 'D,' 'N'
            - Combination 6 (“pos_rsa_6”): 'R,' 'V,' 'N,' 'V,' 'D,’ 'A'
            - Combination 7 (“pos_rsa_7”): 'R,' 'V,' 'O,' 'V,’ 'D,' 'N'
            - Combination 8 (“pos_rsa_8”): 'R,' 'V,' 'O,' 'V,' 'N,' 'A'
            - Combination 9 (“pos_rsa_9”): 'R,' 'V,' 'D,' 'N,' 'V,' 'D,' 'N'
            - Combination 10 (“pos_rsa_10”): 'R,’ ', 'V,', 'O,', 'V,', 'P,'', 'D,', 'N'
            - Combination 11 (“pos_rsa_11”): 'R,’ 'V,' 'O,' 'V,' 'D,' 'A,' 'N'
            - Combination 12 (“pos_rsa_12”): 'R,' 'V,' '^,' ‘^,’ 'V,' 'R,' 'P'

    - Token-related:
        - Approximate negator combined with tokens from the “any” word group;
- Features from annotation recorded observations:
    - Approximate negator combined with specific tokens (bigrams or trigrams):
        - “barely” ngrams: “barely_even,” “barely_ever,” “barely_ enough,” “barely_at_all;”
        - “hardly” ngrams: “hardly_even,” “hardly_ever;”
        - “rarely” ngrams: “rarely_if_ever,” “rarely_even,” “very_ rarely;”
        - “scarcely” ngrams: “scarcely_at_all,” “scarcely_if_ever,” “scarcely_enough;”
        - “seldom” ngrams: “very_seldom,” “seldom_if_ever,” “but_seldom.”

Results by approximate negator. The following tables show the classifier’s performance by sub-corpora, according to each specific approximate negator. As

[16] Description of ARK tags (Penn tags offered between brackets for reference): ‘A’: adjective (J*); ‘D’: determiner (WDT, DT, WP$, PRP$); ‘N’: common noun (NN, NNS); ‘V’: verb (including copula & auxiliaries (V*, MD); ‘R’: adverb (R*, WRB); ‘O’: pronoun (personal/WH; not possessive; PRP, WP); ‘^’: proper noun (NNP, NNPS).

explained in Chapter 3, we used 5-fold cross validation for precision and recall in all of our experiments.

Each table shows both the classifier and confusion matrix result values for each approximate negator group. As also explained in Chapter 3, classifier score values offered are precision, recall, and F1, while confusion matrix figures refer to true positives, false positives, and false negatives (for error analysis). Since the focus of this research is on full reversal detection, the table shows classifier prediction values for the Full negation label first, followed by the average for all labels (Partial, Ambiguous, and Full), and finally the confusion matrix values for true positive (Full tweets predicted as Full), false positive (Ambiguous and Partial tweets labeled as Full) and false negative (Full tweets predicted as Ambiguous or Partial).

For each approximate negator, two tables are offered (named with sub-indexes A and B). Starting with a baseline consisting of the most frequent 2,000 terms, Table A shows aggregated performance values for features, i.e. each feature is added to the one before while they improve performance, and those features that do not make improvements are turned off. Table B shows the performance of each feature individually on the baseline in a non-aggregated way. For each approximate negator, the number of bigrams and trigrams giving the best performance are offered along with the total number of bigrams and trigrams with minimum occurrence of 3 (noted between parenthesis). Finally, the best performing combinations of part of speech tags for reverse subject auxiliary detection (“pos_rsa” values) are also offered. For instance, in the case of “barely,” all bigrams helped improve performance (280 modeled out of 280 total bigrams for this word), while a value of 30 for trigrams (out of a total of 65) helped raise the performance. Regarding reversed subject-auxiliary, a combination of part of speech tags 2, 3, 4, and 5 showed the best scores. As shown above, these pos_rsa combinations are:

```
pos_rsa_2 = ['R,' 'V,’ 'N,' 'V,' 'N']
pos_rsa_3 = ['R,' 'V,' 'O,' 'V']
pos_rsa_4 = ['R,' 'V,' 'O,' 'V,' 'N']
pos_rsa_5 = ['R,' 'V,' 'N,' 'V,' 'D,' 'N']
```

During the discussion of the experiment results, the Greek letter delta ("δ") represents the difference between two values, typically an increase over the baseline after a particular feature or set of features is added.

- ***"Barely"***

Table 18.A: Negation detection: Aggregated features performance for "barely":

Measure	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
Feature	Prec[17]	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.51	0.51	0.51	0.81	0.81	0.81	125	121	118	929
Bigrams, 280 (280 total)	0.51	0.53	0.52	0.81	0.81	0.81	129	124	114	925
Trigrams, 30 (65 total)	0.51	0.52	0.51	0.81	0.81	0.81	126	121	117	928
Num_hashtags	0.51	0.52	0.52	0.81	0.81	0.81	127	121	116	928
At_mentions	0.51	0.52	0.51	0.81	0.81	0.81	127	124	116	926
Barely_any	0.66	0.71	0.68	0.87	0.87	0.87	172	88	71	960
Barely_even	0.71	0.71	0.71	0.88	0.88	0.88	173	72	70	977
Barely_ever	0.71	0.72	0.71	0.88	0.89	0.88	174	70	69	979
Barely_enough	0.71	0.72	0.71	0.88	0.89	0.88	174	70	69	979
Barely_at_all	0.73	0.74	0.73	0.89	0.89	0.89	179	67	64	982
Reversed_polarity_tag	0.72	0.73	0.73	0.89	0.89	0.89	177	68	66	981
Reversed_subject_auxiliary (pos_rsa 2,3,4,5)	0.73	0.73	0.73	0.89	0.89	0.89	178	65	65	983

[17] Abreviations for all tables: "prec" for "precision"; "rec" for "recall"; "avg" for "average"; "pos" for positive; "neg" for negative.

Table 18.B: Negation detection: Individual feature performance for “barely”:

Measure	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.51	0.51	0.51	0.81	0.81	0.81	125	121	118	929
Bigrams, 280 (280 total)	0.51	0.53	0.52	0.81	0.81	0.81	129	124	114	925
Trigrams, 30 (65 total)	0.51	0.50	0.50	0.80	0.81	0.81	122	119	121	930
Num_hashtags	0.50	0.51	0.50	0.80	0.81	0.81	123	123	120	927
At_mentions	0.52	0.52	0.52	0.81	0.81	0.81	127	118	116	932
Barely_any	0.65	0.70	0.67	0.86	0.87	0.87	169	91	74	958
Barely_even	0.58	0.55	0.56	0.83	0.84	0.83	133	95	110	954
Barely_ever	0.50	0.52	0.51	0.81	0.81	0.81	127	126	116	924
Barely_enough	0.50	0.52	0.51	0.81	0.81	0.81	126	124	117	926
Barely_at_all	0.52	0.53	0.53	0.81	0.82	0.81	129	117	114	932
Reversed_polarity_tag	0.50	0.51	0.51	0.80	0.81	0.81	124	122	119	928
Reversed_subject_auxiliary (pos_rsa 2,3,4,5)	0.51	0.51	0.51	0.80	0.81	0.81	124	118	119	932

Regarding aggregated features results (Table 1), the number of hashtags and bigrams are the lexical features improving the performance of Full negation prediction only slightly, from an F1 score of 0.51 to 0.52; however, the all-labels average stays steady at 0.81 F1 (the same as the baseline). Regarding features specific to this research, “barely_any” shows the first boost in aggregated features performance, with F1 values going from 0.51 to 0.68 (δ = 0.17) in Full negation detection scores and from 0.81 to 0.87 (δ = 0.06) in the all-labels average results. The positive impact of this feature on performance is seen in both the aggregated values and the individual values as well, with exactly the same values for all-labels average F1 scores and showing the highest score for Full negation (0.67 as a disaggregated value). Additionally, aggregating this feature with the standard lexical ones also improves error handling, with true positives in the confusion matrix bursting up from 127 to 172; false positive and false negative values going down (124 to 88 and 116 to 71 respectively); and finally, true negatives also increasing, from 926 to 960, confirming the improved performance of the classifier. This improvement in effectiveness is also replicated in the disaggregated table, with true positives going up from 125 to 169 in the baseline, false positives decreasing from 121 to 91, and false negatives decreasing from 118 to 74, which shows the importance of this feature alone in training the classifier to automatically detect negated tweets. The next highest performance improvement is offered by the "barely_even" feature. Focusing on the aggregated values table, this feature shows an improvement from 0.68 to 0.71 (δ = 0.03) in the Full prediction F1 score and from 0.87 to 0.88 (δ = 0.01) in all-labels average F1 values. When analyzing individual values, scores for this feature go from 0.51 to 0.56 (δ = 0.05) for Full F1, and from 0.81 to 0.83 for the all-labels average (δ = 0.02). After that, in the aggregated experiments, "barely_at_all" raises the Full F1 score to 0.73, while the all-labels average reaches 0.89 (δ = 0.01), also increasing true positive and negative values while lowering false values and thus decreasing errors. In the individual scores table, "barely_at_all" shows increased values from the baseline for Full negation F1 scores only, with 0.53 (0.51 baseline). For the aggregated experiments, F1 values stay steady at those values in spite of added features, which nevertheless improve performance in the confusion matrix table, with final scores of 178 for true positives (baseline at 125); 65 for both false positives (baseline at 121) and negatives (baseline: 118); and true negatives of 983 (with baseline 929).

As mentioned previously, the best performance regarding reverse subject auxiliary part of speech tags involved a combination of verbs, nouns, and pronouns headed by and adverb, the approximate negator itself, as follows:

- Adverb, verb, noun, verb, noun (POS combination: 'R,' 'V,' 'N,' 'V,' 'N;' feature set up: pos_rsa_2);
- Adverb, verb, pronoun, verb (POS combination: 'R,' 'V,' 'O,' 'V;' feature set up: pos_rsa_3);
- Adverb, verb, pronoun, verb, noun (POS combination: 'R,' 'V,' 'O,' 'V,' 'N;' feature set up: pos_rsa_4);
- Adverb, verb, noun, verb, determiner, noun (POS combination: 'R,' 'V,' 'N,' 'V,' 'D,' 'N;' feature set up: pos_rsa_5)

Overall, we see that all features helped improve performance for "barely." This finding, however, will not apply to the rest of the approximate negator groups, in which some features cancel each other out or become irrelevant to boosting the scores.

- ***"Hardly"***

Table 19.A: Negation detection: Aggregated features performance for "hardly":

Measure	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.69	0.66	0.67	0.69	0.69	0.69	406	184	213	495
Bigrams, 0 (331 total)	----	----	----	----	----	----	----	----	----	----
Trigrams, 0 (113 total)	----	----	----	----	----	----	----	----	----	----
Num_hashtags	----	----	----	----	----	----	----	----	----	----
At_mentions	----	----	----	----	----	----	----	----	----	----
Hardly_any	0.76	0.71	0.73	0.76	0.76	0.75	438	135	181	544
Hardly_even	0.77	0.71	0.74	0.76	0.76	0.76	439	133	180	546
Hardly_ever	0.77	0.71	0.74	0.76	0.76	0.76	440	134	179	545
Reversed_polarity_tag	0.77	0.71	0.74	0.76	0.76	0.76	441	133	178	546
Reversed_subject_auxiliary	----	----	----	----	----	----	----	----	----	----

Table 19.B: Negation detection: Individual feature performance for "hardly":

Measure	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.69	0.66	0.67	0.69	0.69	0.69	406	184	213	495
Bigrams, 0 (331 total)	----	----	----	----	----	----	----	----	----	----
Trigrams, 0 (113 total)	----	----	----	----	----	----	----	----	----	----
Num_hashtags	0.69	0.66	0.67	0.69	0.69	0.69	406	182	213	497
At_mentions	0.69	0.66	0.67	0.69	0.69	0.69	406	181	213	497
Hardly_any	0.77	0.71	0.74	0.76	0.76	0.76	438	134	181	545
Hardly_even	0.69	0.66	0.68	0.70	0.70	0.70	408	181	211	498
Hardly_ever	0.69	0.66	0.67	0.69	0.70	0.69	408	183	211	496
Reversed_polarity_tag	0.69	0.66	0.67	0.69	0.69	0.69	407	183	212	496
Reversed_subject_auxiliary	----	----	----	----	----	----	----	----	----	----

In the case of the approximate negator “hardly,” neither bigrams or trigrams improved the performance of the classifier for both aggregated and individual tables. The next considerable improvement occurs when introducing “hardly_any.” In the aggregated scores table, we see a boost from 0.67 to 0.73 in the case of Full negation prediction, and from 0.69 to 0.75 regarding all-labels average F1 (δ = 0.06 in both cases). In both aggregated and individual experiments, “hardly_even” slightly increases the F1 scores, by only δ = 0.01; in the case of aggregated results, it shows 0.74 for Full negation prediction and 0.76 for all-labels average, while the individual shows 0.68 (baseline: 0.67) and 0.70 (baseline: 0.69) respectively. Although they do not enhance F1 scores, they do improve confusion matrix values, raising the true positives up to 441 (out of baseline: 406) and true negatives to 546 (baseline: 495) as well as lowering falsely predicted values (from 184 in the baseline to 133 and false negatives from 213 to 178) and raising true negatives from 495 to 546; all of these confusion matrix scores highlight the improved performance of the classifier. Finally, reverse subject auxiliary part of speech tags had no effect on the scores either and were turned off during the experiments.

- ***“Rarely”***

Table 20.A: Negation detection: Aggregated features performance for “rarely”:

Measure	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.51	0.53	0.52	0.78	0.78	0.78	151	148	136	859
Bigrams, 200 (311 total)	----	----	----	----	----	----	----	----	----	----
Trigrams, 73 (73 total)	----	----	----	----	----	----	----	----	----	----
Num_hashtags	----	----	----	----	----	----	----	----	----	----
At_mentions	----	----	----	----	----	----	----	----	----	----
Rarely_any	0.56	0.57	0.57	0.80	0.80	0.80	165	129	122	877
Rarely_even	0.56	0.59	0.57	0.80	0.80	0.80	168	131	119	875
Rarely_ever	----	----	----	----	----	----	----	----	----	----
Rarely_if_ever	0.57	0.60	0.58	0.81	0.80	0.81	171	130	116	875
Very_rarely	0.70	0.70	0.70	0.86	0.86	0.86	202	85	85	922
V_rarely	0.72	0.70	0.71	0.87	0.87	0.87	202	80	85	928
reverse_polarity_tag	----	----	----	----	----	----	----	----	----	----
reverse_subject_auxiliary (pos rsa: 1, 2, 4, 6, 11)	0.73	0.73	0.73	0.87	0.88	0.88	209	76	78	931

Table 20.B: Negation detection: Individual feature performance for “rarely”:

Measure	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.51	0.53	0.52	0.78	0.78	0.78	151	148	136	859
Bigrams, 200 (311 total)	0.50	0.53	0.51	0.78	0.77	0.77	151	152	136	854
Trigrams, 73 (73 total)	0.53	0.53	0.51	0.78	0.77	0.77	151	154	135	853
Num_hashtags	0.49	0.51	0.50	0.77	0.77	0.77	147	154	140	852
At_mentions	0.50	0.52	0.51	0.77	0.77	0.77	149	152	138	854
Rarely_any	0.56	0.58	0.57	0.80	0.80	0.80	166	132	121	874
Rarely_even	0.51	0.54	0.53	0.78	0.78	0.78	155	146	132	860
Rarely_ever	0.49	0.51	0.50	0.77	0.77	0.77	147	153	140	853
Rarely_if_ever	0.50	0.53	0.52	0.78	0.78	0.78	153	151	134	857
Very_rarely	0.65	0.67	0.66	0.84	0.84	0.84	191	102	96	906
V_rarely	0.50	0.53	0.51	0.78	0.77	0.77	151	153	136	854
reverse_polarity_tag	0.49	0.53	0.51	0.77	0.77	0.77	151	155	136	852
reverse_subject_auxiliary (pos rsa: 1, 2, 4, 6, 11)	0.52	0.55	0.53	0.78	0.78	0.78	157	147	130	859

For “rarely” sub-corpus aggregated experiments, features “v_rarely” and “very_rarely” contributed the higher accumulative increase in F1 score values, with 0.71

in Full negation prediction (δ = 0.12) and 0.87 in all-labels average (δ = 0.06) for “v_rarely,” after reaching 0.70 with “very_rarely” in Full negation and 0.86 for all-labels average F1 scores. These features also significantly improve the confusion matrix values, increasing true positives from 171 to 202, lowering false positives and negatives to 85 (from 130 and 116 respectively) and raising true negatives from 875 to 928. The final values for the aggregated table shows F1 values of 0.73 for Full negation prediction (0.52 at the baseline, δ = 0.21) and 0.88 for the all-labels average (0.78 at the baseline, δ = 0.10).

However, feature performance for the individual experiments behaves differently. Although “very_rarely” still offers the most significant improvement from 0.52 in the baseline to 0.66 (δ = 0.14) for Full negation F1 scores and from 0.78 at the baseline to 0.84 (δ = 0.06) in the case of the all-labels average, the next considerable feature in F1 increase is “rarely_any,” reaching 0.57 from the Full negation prediction baseline (δ = 0.05), and up to 0.80 (δ = 0.02) for the all-labels average. The rest of the features either lower the baseline or have negligible improvement values.

Finally, the specific combination of part of speech tags that improved performance are as follows:

- Adverb, verb, pronoun, verb, pronoun (POS combination: 'R,' 'V,' 'O,' 'V,' 'O;' feature set up: pos_rsa_1);
- Adverb, verb, noun, verb, noun (POS combination: 'R,' 'V,' 'N,' 'V,' 'N;’ feature set up: pos_rsa_2);
- Adverb, verb, pronoun, verb, noun (POS combination: 'R,' 'V,' 'O,' 'V,' 'N;' feature set up: pos_rsa_4);
- Adverb, verb, noun, verb, determiner, adjective (POS combination: 'R,' 'V,' 'N,' 'V,' 'D,' 'A;' feature set up: pos_rsa_6);
- Adverb, verb, pronoun, verb, determiner, adjective, noun (POS combination: 'R', 'V,' 'O,' 'V,' 'D,' 'A,' 'N;' feature set up: pos_rsa11)

- ***"Scarcely"***

Table 21.A: Negation detection: Aggregated features performance for "scarcely":

Measure Feature	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.59	0.54	0.57	0.70	0.71	0.70	176	120	147	741
Bigrams, 200 (327 total)	----	----	----	----	----	----	----	----	----	----
Trigrams, 99 (99 total)	----	----	----	----	----	----	----	----	----	----
Num_hashtags	----	----	----	----	----	----	----	----	----	----
At_mentions	----	----	----	----	----	----	----	----	----	----
Scarcely_any	0.62	0.56	0.59	0.71	0.71	0.71	182	113	141	746
Scarcely_at_all	----	----	----	----	----	----	----	----	----	----
Scarcely_ever	0.61	0.57	0.59	0.71	0.71	0.71	184	118	139	739
Scarcely_if_ever	----	----	----	----	----	----	----	----	----	----
Scarcely_enough	----	----	----	----	----	----	----	----	----	----
Reversed_polarity_tag	----	----	----	----	----	----	----	----	----	----
reverse_subject_auxiliary	----	----	----	----	----	----	----	----	----	----

Table 21.B: Negation detection: Individual feature performance for "scarcely":

Measure Feature	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.59	0.54	0.57	0.70	0.71	0.70	176	120	147	741
Bigrams, 200 (327 total)	0.59	0.54	0.57	0.70	0.70	0.70	176	121	147	740
Trigrams, 99 (99 total)	0.59	0.55	0.57	0.70	0.70	0.70	178	124	145	735
Num_hashtags	0.60	0.55	0.57	0.70	0.71	0.70	177	117	146	742
At_mentions	0.60	0.54	0.57	0.70	0.71	0.70	176	117	147	743
Scarcely_any	0.61	0.56	0.59	0.71	0.71	0.71	182	115	141	745
Scarcely_at_all	0.59	0.54	0.57	0.70	0.70	0.70	175	120	148	741
Scarcely_ever	0.59	0.55	0.57	0.70	0.70	0.70	177	121	146	737
Scarcely_if_ever	0.60	0.54	0.57	0.70	0.70	0.70	174	118	149	742
Scarcely_enough	0.60	0.55	0.58	0.71	0.71	0.71	178	117	145	745
Reversed_polarity_tag	0.60	0.55	0.57	0.70	0.71	0.70	177	120	146	742
reverse_subject_auxiliary	----	----	----	----	----	----	----	----	----	----

In the case of "scarcely," only two features prove to be relevant for raising F1 scores: "scarcely_any" and "scarcely_ever." The Full negation F1 performance for both features is the same in the aggregated and individual experiments, from 0.57 to 0.59 (δ = 0.02); while the all-labels average F1 slightly, from 0.70 to 0.71 (δ = 0.01). Regarding confusion matrix values, the best performance in true positives is given by “scarcely_ever” with an increase from 176 to 184 (aggregated values), while “scarcely_any” contributes the best results for true negatives with 746 at a baseline of 741. In the case of individual experiments, “scarcely_any” again reports a slight increase in F1 scores, from 0.57 in the baseline to 0.59 in the case of Full negation prediction, and from 0.70 to 0.71 for the all-labels average; “scarcely_enough” follows with another slight increase. The individual experiment F1 values for the remaining features stay at the baseline.

It is worth mentioning that “scarcely” is, by far, the sub-corpus that registered the largest number of tweets labeled as Ambiguous with 31.6%; in contrast, “seldom” offers the second largest percentage of Ambiguous tweets in a sub-corpus, with only 2.5%. We see that there is a 29.1% gap between these scores, which shows the noisy data composition of the sub-corpus “scarcely.” The following table shows the percentage of Ambiguous for each sub-corpus:

Sub-corpus	# labeled Ambiguous	Percentage
Barely	8	0.6
Hardly	2	0.1
Rarely	5	0.3
Scarcely	443	31.6
Seldom	35	2.5

Table 22: Percentage of tweets labeled “Ambiguous” in each sub-corpus

As discussed in the “Twitter data related issues” section (Chapter 3), this bias towards the Ambiguous label in the sub-corpus “scarcely” relates to the high number of robot-generated tweets in this particular group, with 45.1% of its tweets generated by bot URLs. Undoubtedly, this high number of "noisy" data contributed to the low performance score of the classifier in the "scarcely" sub-corpus.

- ***"Seldom"***

Table 23.A: Negation detection: Aggregated features performance for "seldom":

Measure Feature	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.46	0.48	0.47	0.74	0.73	0.73	140	162	153	809
Bigrams, 400 (499 total)	0.47	0.48	0.48	0.74	0.73	0.74	141	157	152	812
Trigrams, 30 (190 total)	----	----	----	----	----	----	----	----	----	----
Num_hashtags	----	----	----	----	----	----	----	----	----	----
At_mentions	----	----	----	----	----	----	----	----	----	----
Seldom_any	0.49	0.50	0.56	0.75	0.74	0.74	146	149	147	820
Seldom_ever	0.50	0.50	0.50	0.75	0.74	0.75	147	149	146	821
Seldom_if_ever	0.50	0.51	0.51	0.75	0.75	0.75	149	148	144	824
But_seldom	0.53	0.54	0.54	0.76	0.76	0.76	159	142	134	828
Very_seldom	0.63	0.66	0.65	0.81	0.81	0.81	193	112	100	855
v_seldom	----	----	----	----	----	----	----	----	----	----
Reversed_polarity_tag	----	----	----	----	----	----	----	----	----	----
Reversed_subject_auxiliary (pos rsa: 1, 2, 3)	0.64	0.65	0.65	0.81	0.81	0.81	191	108	102	861

Table 23.B: Negation detection: Individual feature performance for "seldom":

Measure Feature	Full negation label values			Average values (all labels)			Confusion matrix for Full negation labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Baseline: words	0.46	0.48	0.47	0.74	0.73	0.73	140	162	153	809
Bigrams, 400 (499 total)	0.47	0.48	0.48	0.74	0.73	0.74	141	157	152	812
Trigrams, 30 (190 total)	0.46	0.47	0.47	0.73	0.73	0.73	139	163	154	805
Num_hashtags	0.47	0.48	0.48	0.74	0.73	0.74	141	156	152	814
At_mentions	0.47	0.48	0.47	0.74	0.73	0.73	140	160	153	811
Seldom_any	0.49	0.50	0.49	0.75	0.74	0.75	146	151	147	821
Seldom_ever	0.47	0.48	0.48	0.74	0.73	0.74	141	157	151	814
Seldom_if_ever	0.48	0.48	0.48	0.74	0.74	0.74	140	154	153	818
But_seldom	0.52	0.53	0.52	0.76	0.76	0.76	155	143	138	827
Very_seldom	0.61	0.61	0.61	0.80	0.79	0.80	180	115	113	852
v_seldom	0.48	0.48	0.48	0.74	0.74	0.74	142	153	151	817
Reversed_polarity_tag	0.47	0.48	0.48	0.74	0.73	0.74	141	159	152	814
Reversed_subject_auxiliary (pos rsa: 1, 2, 3)	0.49	0.49	0.49	0.74	0.74	0.74	144	150	149	820

From the lexical features tested, only bigrams slightly contributed to raise F1 scores in both tables, from 0.47 to 0.48 in the case of Full negation prediction and 0.73 to 0.74 for all-labels average (δ = 0.01 in each case). Regarding specific approximate negation features, “very_seldom” shows the most significant score increase, with the aggregated table showing a significant rise from 0.54 (at the last feature, “but_seldom”) to 0.65 (δ = 0.11) in Full negation prediction and from 0.76 (at “but_seldom”) to 0.81 (δ = 0.05) for all-labels average F1. Meanwhile, the confusion matrix presents an improvement from 159 true positives to 193 and true negatives going from 828 to 855 and a decrease of both false positives (from 142 to 112) and false negatives (134 to 100), which indicates lower error rates. Regarding individual feature scores, the table displays an increase from 0.47 in the baseline to 0.61 F1 (δ = 0.14) for Full negation prediction, while the all-labels average rises from 0.73 to 0.80 (δ = 0.07); the confusion matrix shows true positive values going up from a baseline of 140 to 180 and true negatives from 809 to 852, as well as false positives from 162 to 115 and false negatives from 153 to 113, all indicators of better error handling and more accurate prediction.

The next features enhancing score values are “but_seldom” and “seldom_any.” For “but_seldom,” the aggregated F1 for Full negation increases from 0.51 at “seldom_if_ever” to 0.54 (δ = 0.03), but the all-labels average F1 improves only slightly, from 0.75 to 0.76. Aggregated F1 values for “seldom_any” vary between a Full negation and an all-labels average, increasing the score to 0.56 (0.48 at bigrams, δ = 0.08) for the former, while the all-labels average stays steady to 0.74. For the individual experiments, with a baseline of 0.47, these features raise scores to 0.52 and 0.49

values respectively in Full negation F1, while for the all-labels average and with a baseline of 0.73, their scores are 0.76 for “but_seldom” (δ = 0.03) and 0.75 (δ = 0.02) respectively.

Finally, the reversed subject-auxiliary feature did not improve performance in the aggregated experiments, but it raised F1 scores in the individual feature ones, although modestly: for Full negation from 0.47 in the baseline to 0.49 (δ = 0.02) and from 0.73 to 0.74 (δ = 0.01) for the all-labels average. The following are the POS combinations selected for this feature:

- Adverb, verb, pronoun, verb, pronoun (POS combination: 'R,' 'V,' 'O,' 'V,' 'O;' feature set up: pos_rsa_1);
- Adverb, verb, noun, verb, noun (POS combination: 'R,' 'V,' 'N,' 'V,' 'N;’ feature set up: pos_rsa_2);
- Adverb, verb, pronoun, verb (POS combination: 'R,' 'V,' 'O,' 'V,' feature set up: pos_rsa_3).

Overall Findings for Automatic Detection of Negation.

Individual feature performance. Under each approximate negation result section, Table 2 focuses on the individual impact of distinctive features in the classifier’s performance. This section briefly discusses overall findings regarding such an impact. Aggregated feature experiments are will be discussed afterwards.

Lexical and Twitter-related features**.** Overall (and except for “barely” and “seldom”), bigrams, trigrams, number of hashtags and at-mentions seemed to play a negligible role or no role at all in improving the F1 scores for both Full negation and all-labels prediction F1. In most cases though, results go up slightly in the confusion matrix section. Consequently, these standard features could be ignored when predicting approximate negators behaving as full reversal shifters.

Features derived from the pragmatic linguistics literature review. As explained in the discussion of the results of each approximate negator, all “any” features (“barely_any,” “hardly_any,” “rarely_any,” “scarcely_any,” and “seldom_any”) contributed to improve F1 performance in both aggregated and individual feature experiments. Consequently, these experiments confirm findings from the literature indicating that the presence of the approximate negator with a word from the “any” family can be modeled to predict full negation automatically in tweets.

The following table summarizes results for all “any” related features, ranked by their positive effect on Full negation label prediction results (from higher to lower), detailing also all-labels average performance scores:

Measure	Full negation label prediction			All-labels average		
Feature	Baseline	Results	Improvement (δ)	Baseline	Results	Improvement (δ)
“barely any”	0.51	0.67	0.16	0.81	0.87	0.05
“hardly any”	0.67	0.74	0.07	0.69	0.76	0.07
“rarely any”	0.52	0.57	0.05	0.78	0.80	0.02
“seldom any”	0.47	0.49	0.02	0.73	0.75	0.02
“scarcely any”	0.57	0.59	0.02	0.70	0.71	0.01

Table 24: F1 results for “any” related features in “Full” negation label and all-labels prediction tasks

We can see that “barely_any” offers the best result when training a classifier to predict negation, with δ = 0.16, while “hardly_any” has better effect on all-labels prediction with δ = 0.07 in the all-labels average, with “barely_any” in second place with δ = 0.05. Moreover, “barely_any” has a remarkable improvement in Full negation prediction scores in comparison with the rest of the sub-corpora experiments for the same “any-“ feature family, the second-best performing being “hardly_any” at δ = 0.07 (δ = 9 lower); thus, these experiments help us conclude that the NPI “any” combined with “barely” helps in predicting negated tweets better than the rest of the approximate negators. On the other hand, “any” combined with “hardly” offers similar prediction effectiveness for negation detection (0.74 F1) than do the rest of the labels (0.71 F1) with only a δ = 0.03, while the gap between negation and all-labels prediction in the rest of the cases generally hovers around δ = 0.20 (δ = 0.20 for barely; δ = 0.23 for rarely; δ = 0.26 for seldom) except for “any” with “scarcely” at δ = 0.12. In consequence, “any” as a polarity-sensitive item of “hardly” does not seem to support the prediction of negative polarity with more effectiveness than with partial polarity.

A caveat related to these findings is the fact that words from the family “any” can also function as free-choice items, as explained in the qualitative analysis results discussion. In this usage, “any” becomes a polarity-neutral item indicating that a property of action can arbitrarily be applied to some member of a class. In the example: “Jan will read almost any computer magazine,” the use of “any” indicates that Jan’s choices of computer magazines apply to members of that class indistinctly. This type of usage shows in tweets as well, which could have an impact on the low performance of the “any” set of features when signaling full negation. For example, the use of the pronoun “anyone” in the tweet “Anyone who is able should be taking a nap right now. Conditions are rarely this perfect” (RA456) indicates that any member of the tweet’s recipient group can take the action suggested by the Twitter user. Thus, there is no negation in this case. The use of “any” as free choice item could be explored in future research to develop features that will differentiate both phenomena more clearly.

Conversely to the case of “any,” other features found in the literature seemed to have a lower impact on machine learning score improvement. Two cases stand out: reversed polarity tags and reverse subject-auxiliary features. Regarding reversed polarity tags, “barely” and “hardly” were the only subsets in which this feature contributed to some improvement, though it was only marginal and was reduced to a few tweets gained as true positives (or lowering the “false” values) in the confusion matrix results.

The reversed subject-auxiliary feature made a significant contribution only in the case of “seldom,” increasing the Full negation prediction scores for $\delta = 0.2$ (from 0.47 to 0.49) and the all-labels average results slightly for $\delta = 0.01$ (from 0.73 to 0.74). For “rarely,” this feature also improved Full negation scores by $\delta = 0.01$, from 0.52 to 0.53 F, while all-labels F1 performance stayed steady at the baseline (0.78). Finally, in the case of “barely,” this feature didn’t present any improvement, while in the case of “hardly” and “scarcely” it actually lowered performance and had to be turned off. Consequently, according to this study, the reversed subject-auxiliary style of negation presents a better chance of signaling full negation in tweets including the word “seldom” primarily, and with less chances for “rarely” than in any other cases in the corpus.

Neither reversed subject-auxiliary nor reversed polarity tag features contributed to increase the performance of the classifier for the “scarcely” sub-corpus; however, we should remember that, in general, this token showed the highest number of noisy data in the corpus due to bot activity. Moreover, this subset showed the lowest impact of features in performance, with only 3 out of 11 features affecting the classifier scores.

A caveat about the effectiveness of these features relates to feature overlap. The following tweet contains tokens that could have been picked up by two types of features: the expression “very_rarely” could had been detected by the specific feature that names it, and the tokens “rarely do I call out of work” could have also been found by the pos_rsa 10 combination of the reversed subject auxiliary feature (i.e., the part of speech combination “adverb, verb, pronoun, verb, preposition or conjunction, determiner, noun” or “'R,' 'V,' 'O,' 'V,' 'P,' 'D,' 'N'” in POS tags):

- “Very rarely do I call out of work or make excuses for myself, but today is just one of those days” (RA280)

Another example along the same lines:

- “Very seldom do I consider friends like family. But those that I do, know they're family.” (SE378)

***Features that emerged from human annotation*.** Observation notes during the human annotation phase helped create a number of features that are specific to each approximate negation adverb. As analyzed in the previous sections, those features proved to yield relevant results for the machine learning process in all cases except for the “scarcely” sub-corpus, presumably due to the high amount of noisy data coming from bots.

The following list (with values extracted from each Table B in the “Results by approximate negator” section) shows these features ranked by their impact on the all-labels average F1 score. Features with no impact on the overall F1 results are excluded:

Measure / Feature	Full negation label prediction			All-labels average		
	Baseline	Results	Improvement (δ)	Baseline	Results	Improvement (δ)
“very_seldom”	0.47	0.61	0.14	0.73	0.80	0.07
“very_rarely”	0.52	0.66	0.14	0.78	0.84	0.06
“barely_even”	0.51	0.56	0.05	0.81	0.83	0.02

“but_seldom”	0.47	0.52	0.05	0.73	0.76	0.03
“barely_at_all”	0.51	0.53	0.02	0.81	0.81	0.00
“hardly_even”	0.67	0.68	0.01	0.69	0.70	0.01
“scarcely_enough”	0.57	0.58	0.01	0.70	0.71	0.01
“seldom_ever”	0.47	0.48	0.01	0.73	0.74	0.01
“seldom_if_ever”	0.47	0.48	0.01	0.73	0.74	0.01
“v_seldom”	0.47	0.48	0.01	0.73	0.74	0.01
“rarely_even”	0.52	0.53	0.01	0.78	0.78	0.00

Table 25: Impact of features emerged from human annotation in F1 scores for both Full negation and all-labels prediction tasks

Regarding full negation prediction, only features “very_seldom” and “very_rarely” have a significant impact on the classifier with δ = 0.14 in both cases. They also help improve overall performance, with deltas of 0.07 and 0.06 respectively. Bigrams “barely_even” and “but_seldom” follow with a significantly lower gap of δ = 0.05 for negation prediction, and δ = 0.02 and δ = 0.03 for all-labels prediction respectively. Trigram “barely at all” follows with modest improvement in full negation, δ = 0.02, and was neutral in all-labels prediction. Consequently, these experiments allow us to conclude that bigrams “very seldom” and “very rarely” both help signal negation usage in tweets and also improve the performance of the classifier overall.

Regarding the rest of the features that emerged from annotation, although these linguistic occurrences were considered recurrent and worth analyzing during the human annotation phase, when modeled for machine learning they proved not to be impactful for the classifier’s performance effectiveness. However, and as discussed previously, this could also be explained by the presence of overlapping features; expressions such as “very_rarely ever” in the following tweet could had been picked up by two features, “very_rarely” and “rarely_ever:”

- “@GHmagazine @TheVintageYear I preferred the no-make-up looks. But then I have very rarely ever worn make-up since i was 16” (RA1310)

Exploring these overlaps as well as refining the definition of features to better train the classifier could be the goal of future research endeavors.

Aggregated feature performance: F1 comparison for both Full negation and all-labels average scores. The following table shows the final aggregated values of F1 for all approximate negators:

Measure Feature	Full negation label prediction			All-labels average prediction		
	prec	rec	F1	avg. prec	avg. rec	avg. F1
Barely	0.73	0.73	0.73	0.89	0.89	0.89
Hardly	0.77	0.71	0.74	0.76	0.76	0.76
Rarely	0.73	0.73	0.73	0.87	0.87	0.87
Scarcely	0.61	0.57	0.59	0.71	0.71	0.71
Seldom	0.64	0.65	0.65	0.81	0.81	0.81

Table 26: F1 scores for Full negation label and all-labels prediction tasks

The following is the rank of values, from higher to lower, for both Full negation and all-labels F1 scores:

- Full negation prediction performance: “hardly:” 0.74; “barely” and “rarely:” 0.73; “seldom:” 0.65; “scarcely:” 0.59;
- All-labels average effectiveness: “barely:” 0.89; “rarely:” 0.87; “seldom:” 0.81; “hardly:” 0.76; “scarcely:” 0.71.

Regarding all-labels average F1, “barely” is the best performing approximate negator of all five, with a score of 0.89, followed by “rarely” with 0.87 and “seldom” with 0.81. “Rarely” shows an interesting case of balanced performance, with the even evaluation results in both Full negation (precision, recall and F1 at 0.73) and all-labels average (at 0.87 for all three evaluation measures). “Hardly,” however, offers the most balanced performance across evaluation measures, with all values around the 0.70 range; this approximate negator also shows the highest score in Full negation prediction with 0.74. However, compared to the other words under scrutiny, this adverb drops from top performance in Full negation prediction to next-to-last in all-labels average F1 (just above “scarcely” values, the corpus full of noisy data); this fact could indicate that “hardly” can easily signal full negation, but its other roles in tweets beyond that could be challenging to predict. In any case, the experiments show that “hardly” is a good predictor of Full reversal valence or prototype negation, followed by “barely” and “rarely” also with best overall performance.

An interesting case is offered by “seldom.” Although its prediction score as full negator is the second lowest with 0.65, the all-labels F1 value is considerably high with 0.82, ranked third after “barely” and “rarely.” This fact may indicate that this adverb tends to act as an approximate negator (i.e. producing regular valence shifting) more consistently than as a full negator (or full valence shifter). Consequently, its role in negation prediction could either be dismissed or determined to need further exploration.

Finally, “scarcely” showed the lowest Full negation prediction results with 0.59 and an all-labels average F1 with 0.71, presumably due to the fact that it had the highest number of ambiguous tweets (443, or 31.6% of the sub-corpus) coming from bots.

Confusion matrix values. The following table ranks approximate negators according to the percentage of error shown in the final performance scores. For each sub-corpus, the table offers true positive values along with false positives and false negatives (together under the label “Error”), along with accuracy values. It is worth highlighting that values are offered aggregated for all labels (Full, Partial and Ambiguous) to help in assessing the classifier’s overall performance:

Approximate Negator	Total true positives (all labels)	Total Error	Percentage of error	Accuracy (true positives / total of 1300 tweets per corpus)
Scarcely	925	375	**28%**	71%
Hardly	987	313	**24%**	75%
Seldom	1052	248	**23%**	80%
Rarely	1140	160	**14%**	87%
Barely	1161	139	**12%**	89%

Table 27: Percentage of error in classifier’s performance by sub-corpus

“Scarcely” shows the highest percentage of error with 28% tweets predicted wrong, presumably due to the presence of noisy data. “Seldom” follows with 24%. We can also see that, as accuracy increases, error percentage decreases, which is expected. For these two sub-corpora, we can additionally see that the gap between the number of tweets predicted correctly (in the total true positives column) and those labeled incorrectly is closer than with the rest of the cases. Further studies should be conducted to investigate these errors.

Machine learning vs. human annotation scores. As mentioned in previous discussions, machine learning results are usually compared to human annotation scores (the so-called “human ceiling”) to assess their efficacy. The following table shows inter-annotation Kappa, accuracy, and F1 values for each subset of tweets:

Measure / Sub-corpus	**Human Annotation**		**Machine Learning**	
	Kappa	**Accuracy**		F1
Barely	90.77	97.25%	89.31%	0.89
Hardly	82.40	91.75%	75.92%	0.76
Rarely	91.50	97.25%	87.69%	0.88
Scarcely	80.99	88.00%	71.00%	0.71
Seldom	88.44	95.25%	80.92%	0.82

Table 28: Comparison between machine learning and human annotation scores for automatic detection of negation, by sub-corpus

Although encouraging, several factors limit the interpretation of results in this table. First, the human annotation values shown are calculated based on 400 tweets per approximate negator subset (roughly 31% of each sub-corpus, which correspond to tweets annotated concurrently by both annotators for reliability control during the annotation process), while the machine learning values refer to the entire sample of 1,300 tweets of each corpus. Second, accuracy is highlighted at the center of the table since it is the only type of measure that can be computed for both human annotation and machine learning results. However, as discussed in Chapter 3, accuracy does not account for the different types of errors in classification performance, therefore becoming a limited performance measure compared to F1.

Keeping in mind the aforementioned caveats of sample sizes and measurement type, we can still see some expected trends in the results. Human annotation scores surpass machine learning results but the gap is narrow enough to allow further research on machine learning performance improvement. In particular, “hardly” and “seldom” show the highest gap in accuracy between human and machine results (a 17% and 16% gap in each case), offering opportunities for model improvement.

Discussion. These experiments have employed a number of features for the automatic detection of approximate negators acting as full reverse valence shifters (or prototype negators) in tweets, with the final goal of answering the first part of research sub-question 2. Several features have proven effective for this detection task. Approximate negation words combined with tokens from the “any” family showed better performance when predicting full negation, with remarkable results in the case of “barely.” These were the only findings from the pragmatic linguistic literature that proved themselves to be instrumental to machine learning. From findings originated during human annotation, the expressions “very rarely,” “very seldom,” and “barely even” also consistently signaled the presence of full negation in tweets. Finally, the syntactic feature of reversing a sentence’s subject and its auxiliary verb had a better chance of predicting negation in tweets where the adverb “seldom” is part of the sentence than in any other case.

Comparing the performance of approximate negators with each other, the best performing adverb is “barely” (average F1 of 0.89) and the worst one is “scarcely” (with 0.71 F1), presumably for the presence of noisy data, as discussed. “Hardly” seems to clearly indicate full negation since it offers the highest full negation label prediction with

0.74 F1 for that label; however, its role seems somehow difficult to predict when performing a different shifting valence role, which is indicated by its comparative lower performance in all-labels average F1 value (0.76). “Seldom” presents the opposite case, with better performance as a regular valence shifter (average F1 of 0.81) but lower results for full negation prediction (Full negation prediction F1 of 0.65). “Rarely” stays in between, with consistent higher-than-medium scores for both full and regular valence shifter prediction.

Although there are some limitations in comparing machine learning results to a human ceiling, the values show expected trends and help validate the classifier scores, also showing specific areas of improvement in the model.

Closing remarks: answering the negation prediction component of the second research-sub question. This section presented detailed results for the machine learning experiments conducted in order to answer the first part of the second research sub-question, namely:

> Research sub-question 2: How do we automatically detect approximate negators that behave as reversal valence shifters (or prototype negators) in tweets?

The answer to this question is that approximate negators along with tokens from the family of “any” words in the role of negatively-oriented polarity-sensitive items (NPI) can signal prototype negation in tweets. Additionally, the combination of words “very rarely,” “very seldom” and “barely even” are also good indicators of those approximate negators acting as full reversal shifters. Finally, reversed subject and auxiliary verb syntax will support the role of “seldom” as a prototype negator in Twitter utterances.

The following section will offer the discussion of the experimental results designed to tackle the second component of the answer to the second research sub-question, i.e. once negation is found, how to automatically predict the scope of such negation over other words in the tweet. With this final section, all components of the research question will be answered.

Experiment Results for Automatic Detection of Negation Scope

This automatic detection task was also formulated as a classification task but this time at the token level, i.e. each token in the tweet was labeled as in-scope (I), cue (C), or out-of-scope (O) according to information provided by the set of features described below. It is worth mentioning that, in the case of cue, although in the annotation phase both the approximate negation and the supporting tokens (such as “any” for “barely any”) have been marked as negation cues, in the case of machine learning experiments only the approximate negator’s information (such as its index position in the tweet or part-of-speech tag) was considered for feature engineering. With this approach, we follow Morante and Blanco’s formulation of a partial cue match task (as defined for the *SEM 2012 shared task for resolving the scope of negation) for which only some tokens of the negation cue are detected (Morante & Blanco, 2012; see also Chapter 2 for an extensive discussion of this task).

From the 1,300 tweets for each approximate negator that are part of the corpus, only those labeled as “Full” were processed in this task since only those tweets contain a negation cue for which a scope has to be predicted. As a result, the number of processed tweets (along with tokens labeled) counts as follows: 242 tweets from the “barely” sub-corpus (4,383 tokens classified); 611 from the “hardly” one (11,166 tokens labeled); 286 from “rarely” (5,724 tokens); 321 for “scarcely” (6,388 tokens); and 292 from the “seldom” sub-corpus (5,742 tokens). Finally, as in the case of negation labeling at the tweet level, a 5-fold cross validation was also employed for precision and recall.

Features. Converse to what is typically the case of negation detection, the same features were used for all the approximate negators across the experiments. In each tweet, the approximate negator is referred to as the cue, and features are defined for each token as follows:

Lexical features:

- Token name (baseline for experiments);
- Token POS;
- Bigrams: includes the tokens to the left and right together with the token;
- Trigrams: left and right;
- Bigrams’ POS: includes the POS tags of the tokens to the left and right together with the POS tag of the token;
- Trigram POS: left and right.

Dependency graph related features:

- Token-cue distance: number of tokens between the token and the cue, left and right;
- POS of the first and second heads (for multi-constituent utterances) of the token;
- Position of the token with respect to the cue: either the cue being an ancestor of the token or the token being an ancestor of the cue;
- Dependency path: involving whether or not token and cue share the same head in a dependency graph (true or false), and also the dependency path with regular and reversed arcs.

Both groups of features are labeled with a header in each table. A thorough discussion of these features, along with evaluation measures and other elements of research design can be found in Chapter 3.

As in the case of the full negation detection results reported, for each specific approximate negator two tables are offered: aggregated values in Table A, and individual or disaggregated values in Table B. Evaluation measures also included precision, recall and F1 along with confusion matrix values for error assessment. For these particular experiments, though, since the focus of results evaluation is the prediction of which tokens are under the scope of the negation cue, each table reports the results focusing on in-scope values, leaving cue detection and out-of-scope prediction values out. However, summary tables offer all values for overall assessment performance. As in the previous section, in the discussion of the experiment results, the Greek letter delta ("δ") represents the difference between two values, usually an increase over the baseline after a particular feature or set of features is added.

Results by approximate negator. As in the previous section, aggregated and individual result tables are again provided below.

- ***"Barely"***

Table 29.A: Scope detection: Aggregated features performance for "barely":

Measure Feature	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.38	0.28	0.32	0.51	0.55	0.52	327	531	849	2068
Token POS	0.40	0.31	0.35	0.56	0.56	0.56	363	546	813	2113
Bigrams	0.41	0.31	0.35	0.57	0.56	0.56	364	530	812	2108
Bigrams POS	0.41	0.28	0.33	0.58	0.57	0.57	335	491	841	2169
Trigrams	0.43	0.30	0.35	0.59	0.58	0.58	348	459	828	2212
Trigrams POS	0.44	0.31	0.36	0.59	0.58	0.58	367	472	809	2190
Dependency graph related										
Token-cue distance	0.73	0.74	0.73	0.85	0.84	0.84	873	327	303	2823
First & second head POS	0.74	0.75	0.74	0.85	0.85	0.85	885	318	291	2830
Ancestors (token, cue)	---	---	---	---	---	---	---	---	---	---
Dependency path	0.78	0.77	0.78	0.88	0.88	0.88	911	258	265	2932

For the aggregated results, we see that the token POS feature improves performance by δ = 0.03 in in-scope F1 (from 0.32 to 0.35) and δ = 0.04 in the all-labels average F1 (0.52 to 0.56). Bigrams, trigrams and their POS bump up performance slightly. However, it is the first dependency-related feature, token-cue distance, that raises in-scope F1 scores more than twice with δ = 0.37 (from 0.36 to 0.73). The all-labels average also increased significantly, from 0.58 to 0.84 (δ = 0.26). True positive values go from 367 to 873, which represents more than twice the correctly predicted labels. True negatives increase from 2,190 to 2,823, while false positives and negatives decrease steadily, both indicating better error handling. The second dependency-related feature added, first and second head POS, raises the F1 score by δ = 1 in both values: 0.74 for in-scope and 0.85 for all-labels. After the fourth dependency feature (dependency graph) is aggregated, the final score for in-scope F1 becomes 0.78 (baseline 0.32, δ = 0.46), all-labels average F1 of 0.88 (baseline 0.52, δ =0.36), with 911 true positives (baseline 327) and 2,932 true negatives (baseline 2068), showing a remarkable overall improvement in the performance of the classifier.

Table 29.B: Scope detection: Individual feature performance for “barely”:

Measure Feature	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.38	0.28	0.32	0.51	0.55	0.52	327	531	849	2068
Token POS	0.40	0.31	0.35	0.56	0.56	0.56	363	546	813	2113
Bigrams	0.40	0.29	0.33	0.53	0.55	0.54	338	513	838	2057
Bigrams POS	0.41	0.30	0.35	0.57	0.56	0.56	354	503	822	2113
Trigrams	0.42	0.31	0.36	0.54	0.57	0.55	368	499	808	2110
Trigrams POS	0.41	0.31	0.35	0.55	0.55	0.55	369	534	807	2045
Dependency graph related										
Token-cue distance	0.72	0.73	0.73	0.84	0.84	0.84	861	333	315	2800
First & second head POS	0.41	0.30	0.34	0.56	0.56	0.55	349	511	827	2096
Ancestors (token, cue)	0.60	0.32	0.42	0.59	0.64	0.59	372	243	804	2447
Dependency path	0.73	0.56	0.63	0.82	0.81	0.81	656	247	520	2880

The individual feature experiments show that all features, taken separately, slightly improve scores from the baseline except for token-cue distance and token-cue path. For token-cue distance, there is an abrupt jump of δ = 0.41 from the baseline (from 0.32 to 0.73) for in-scope F1 and an increase of δ = 0.32 in the all-labels average F1 (from 0.52 to 0.84). Additionally, 861 true positive or in-scope values (from 327 in the baseline) and 2,800 true negative (i.e. cue and out-of-scope true positives) values were predicted correctly, with 2,086 as the baseline. The first and second heads POS feature does not raise the baseline significantly, but the dependency path feature introduces the second significant improvement from the baseline (although still slightly lower than token-cue distance), with in-scope F1 0.63 (compared to 0.73 for token-cue distance, 0.32 at the baseline), an all-labels F1 of 0.81 (in comparison to 0.84 in token-cue distance, 0.52 baseline); true positives are 656, significantly lower than token-cue distance values (at 861), but true negatives are almost equal (2,880 and 2,800 respectively) between these two best-performing features. This improvement in true negatives could indicate that the dependency path helped overall performance in predicting true positives for all labels (cue, in-scope, and out-of-scope) but was less effective for in-scope isolated true positives.

- ***"Hardly"***

Table 30.A: Scope detection: Aggregated features performance for "hardly":

Measure	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.38	0.37	0.37	0.55	0.61	0.58	1037	1699	1802	5806
Token POS	---	---	---	---	---	---	---	---	---	---
Bigrams	---	---	---	---	---	---	---	---	---	---
Bigrams POS	---	---	---	---	---	---	---	---	---	---

Trigrams	0.39	0.37	0.38	0.55	0.62	0.58	1044	1664	1795	5891
Trigrams POS	---	---	---	---	---	---	---	---	---	---
Dependency graph related										
Token-cue distance	0.70	0.69	0.69	0.84	0.83	0.83	1947	825	892	7296
First & second head POS	0.70	0.69	0.69	0.84	0.83	0.83	1959	858	880	7279
Ancestors (token, cue)	0.73	0.67	0.70	0.84	0.83	0.83	1907	719	932	7395
Dependency path	0.76	0.71	0.74	0.86	0.86	0.86	2017	630	822	7609

Dependency graph features show the most remarkable impact in performance for the “hardly” approximate negation corpus, starting with token-cue distance reaching 0.69 for in-scope F1 (after trigrams reached 0.37, δ = 0.32), and 0.83 for the all-labels average F1 (0.58 for trigrams, δ = 0.25); almost doubling true positives (from 1,044 for trigrams to 1,947) and true negatives (from trigrams at 5,891 to 7,296); and finally lowering false positives and negatives to almost half (from 1,664 in trigrams to 825, and from 1,802 to 892 respectively). Other dependency-related features modestly continue to improve scores until the last of these features, dependency path, is incorporated; at this point, another substantial increase is present with a raise of the in-scope F1 score to 0.74 (from 0.70), the all-labels F1 to 0.86 (previously 0.83), and although the increase in true positives and true negatives is minor, false positives and negatives decrease significantly (from 719 to 630 and from 932 to 822 respectively), signaling better error treatment. From lexical features, only trigrams increase the baseline for in-scope F1 but slightly, with δ = 0.01 (from 0.37 to 0.38), showing no difference in all-labels F1 and very limited improvement in the confusion matrix values.

Table 30.B: Scope detection: Individual feature performance for “hardly”:

Measure	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.38	0.37	0.37	0.55	0.61	0.58	1037	1699	1802	5806
Token POS	0.36	0.33	0.34	0.55	0.58	0.56	927	1679	1912	5562
Bigrams	0.37	0.35	0.36	0.55	0.61	0.58	996	1672	1846	5807
Bigrams POS	0.37	0.31	0.34	0.57	0.59	0.58	885	1483	1954	5676
Trigrams	0.39	0.37	0.38	0.55	0.62	0.58	1044	1664	1795	5891
Trigrams POS	0.38	0.35	0.36	0.56	0.58	0.57	987	1622	1852	5545
Dependency graph related										
Token-cue distance	0.70	0.67	0.68	0.83	0.82	0.83	1907	831	932	7295
First & second head POS	0.38	0.34	0.36	0.58	0.61	0.59	959	1544	1880	5842
Ancestors (token, cue)	0.57	0.39	0.46	0.61	0.68	0.63	1099	823	1740	6528
Dependency path	0.63	0.49	0.55	0.78	0.77	0.77	1398	814	1441	7180

In the case of the impact of individual features on the baseline, we see again that dependency features tend to train the classifier better. In particular, token-cue distance raises F1 scores from baseline by δ = 0.31 (from 0.37 to 0.68) for in-scope, while the all-labels replicates aggregated values for the same feature, with an increase of δ = 0.25 (from 0.58 to 0.83 as well); true positives almost double from 1,037 at the baseline to 1,907; false positive and negatives are reduced even more effectively (from 1,699 to 831 and from 1,802 to 932 respectively); and true negatives also increase values (from 5,806 at the baseline to 7,297). First and second POS heads do not show significant improvement (actually decreasing the in-scope F1 and true positives values) while ancestors token-cue does show improvement from the baseline, although less than token-cue distance (in-scope F1 with δ = 0.09, from 0.37 to 0.46; and all-labels average with δ = 0.05, from 0.58 to 0.63, and increasing true positives and negatives while lowering false values). Finally, the dependency path feature is second to token-cue distance in effectiveness, with an in-scope F1 value of 0.55 (δ = 0.18, δ = 0.31 for token-cue distance), all-labels average F1 of 0.77 (δ = 0.19, δ = 0.25 for token-cue distance), and raising true positives to 1,398 (1,907 for token-cue distance, baseline at 1,037), slightly lowering false positives but less effective than token-cue distance when it comes to reducing false negatives (from 1,802 at the baseline to 1,441, while token-cue distance reduced the value to 932).

- ***"Rarely"***

Table 31.A: Scope detection: Aggregated features performance for "rarely":

Measure Feature	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.32	0.29	0.31	0.53	0.58	0.55	422	880	1036	2905
Token POS	---	---	---	---	---	---	---	---	---	---
Bigrams	---	---	---	---	---	---	---	---	---	---
Bigrams POS	---	---	---	---	---	---	---	---	---	---
Trigrams	0.39	0.31	0.35	0.56	0.59	0.57	456	719	1002	2924
Trigrams POS	---	---	---	---	---	---	---	---	---	---
Dependency graph related										
Token-cue distance	0.64	0.61	0.62	0.80	0.80	0.80	890	509	568	3689
First & second head POS	---	---	---	---	---	---	---	---	---	---
Ancestors (token, cue)	---	---	---	---	---	---	---	---	---	---
Dependency path	0.67	0.62	0.65	0.82	0.82	0.82	910	448	548	3785

Aggregated features for the sub-corpus "rarely" behave poorly regarding classifier's performance, except for the lexical feature trigrams and the dependency-related features token-cue distance and dependency path. As the individual feature table below shows, other features either didn’t increase or even lowered scores; thus, they were turned off to avoid their negative impact on overall performance. The accumulative effect of the few active features gives a final score of 0.65 for in-scope F1 (baseline of 0.31; δ = 0.34), 0.82 for all-labels average (0.55 value at the baseline; δ = 0.27), and increase in true positives from 422 to 910, decrease of false positives from 880 to 448 while false negatives also go down from 1,036 to 548; finally, true negatives increase to 3,785 from a starting value of 2,905 at the baseline. Token-cue distance is again the feature that brings the first considerable score rise, with δ = 0.31 in the in-scope and δ = 0.27 in the all-labels average F1 values.

Table 31.B: Scope detection: Individual feature performance for “rarely”:

Measure	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
Feature	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.32	0.29	0.31	0.53	0.58	0.55	422	880	1036	2905
Token POS	0.32	0.29	0.31	0.53	0.58	0.55	422	880	1036	2905
Bigrams	0.36	0.30	0.33	0.56	0.59	0.57	444	799	1014	2937
Bigrams POS	0.31	0.20	0.24	0.54	0.56	0.55	291	655	1167	2934
Trigrams	0.39	0.31	0.35	0.56	0.59	0.57	456	719	1002	2924
Trigrams POS	0.34	0.26	0.30	0.54	0.58	0.55	383	728	1075	2918
Dependency graph related										
Token-cue distance	0.60	0.54	0.56	0.78	0.78	0.78	781	531	677	3663
First & second head POS	0.38	0.22	0.28	0.56	0.59	0.57	327	545	1131	3032
Ancestors (token, cue)	0.51	0.22	0.31	0.59	0.61	0.58	328	316	1130	3191
Dependency path	0.58	0.49	0.53	0.76	0.75	0.75	721	528	737	3551

For trigrams, the in-scope F1 score increases from 0.31 at the baseline to 0.35; however, the dependency-related feature token-cue distance presents the highest performance boost with a score of 0.56 (δ = 0.25 from the baseline), while dependency path shows a slightly lower increase 0.53 (δ = 0.22 from baseline) but still higher than lexical features. The all-average F1 also improves with these two features, going from 0.55 at the baseline to 0.78 (δ = 23) for token-cue distance and 0.75 (δ = 20) for dependency path. The confusion matrix shows the same trend, with an increase of true positives from 422 at the baseline to 456 for the lexical feature trigrams, to 781 for the dependency-related feature token-cue distance, and 721 for dependency path, also dependency related. Regarding the rest of the features, the classifier's lowest performance for in-scope prediction comes when adding bigrams POS and with first and second head POS features, with both features generating F1 values lower than the baseline with 0.24 and 0.28 (baseline at 0.31), and also recording the lowest values in true positives with 291 (baseline at 422); this inefficiency in predicting in-scope tokens explains the lower values of in-scope F1. However, the same features perform fairly well in the all-labels F1 score, steadily at 0.55 in the first case, and increasing to 0.57 in the second.

- ***"Scarcely"***

Table 32.A: Scope detection: Aggregated features performance for "scarcely":

Measure Feature	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.53	0.49	0.51	0.55	0.56	0.56	1253	1128	1309	2339
Token POS	0.53	0.52	0.53	0.57	0.57	0.57	1344	1188	1218	2329
Bigrams	0.54	0.55	0.55	0.59	0.59	0.59	1420	1206	1142	2368
Bigrams POS	0.57	0.53	0.55	0.61	0.59	0.60	1349	1038	1213	2451
Trigrams	---	---	---	---	---	---	---	---	---	---
Trigrams POS	0.57	0.53	0.55	0.61	0.60	0.60	1356	1034	1206	2462
Dependency graph related										
Token-cue distance	0.68	0.73	0.70	0.76	0.75	0.75	1873	1086	689	2926
First & second head POS	0.68	0.74	0.71	0.76	0.76	0.76	1890	877	672	2933

Ancestors (token or cue)	0.70	0.72	0.71	0.76	0.76	0.76	1854	799	708	3001
Dependency path	0.76	0.69	0.72	0.79	0.79	0.79	1771	568	791	3252

Except for trigrams, all features showed positive accumulative effects in the classifier’s performance for this approximate negator. However (as seen in previous experiments), the accumulative effect of dependency-related features still boosts scores up more than other types of features. When the first dependency-related feature, token-cue distance, is added to the lexical features, the scores go from 0.55 (trigrams POS) to 0.70 for in-scope F1 (δ = 0.15), and from 0.60 to 0.75 for all-average labels F1 (δ = 0.15), while true positives rising from 1,356 to 1,873 and true negatives from 2,462 to 2,926 (which proves the increase in the effectiveness of the classifier in predicting labels). Interestingly, such effectiveness in label prediction does not apply to the accumulative effect of the latest two features on in-scope true positive values in the confusion matrix, with 1,890 true positives for first and second head POS going down to 1,854 for ancestor token-cue, and finalizing with an even lower value of 1,771; however, the true negative values do increase when those features are added, from 2,933 to 3,001 to a final value of 3,252. Meanwhile, F1 scores also increase at the end, which means that the classifier performs better overall but slightly less so for in-scope token prediction.

Table 32.B: Scope detection: Individual feature performance for “scarcely”:

Measure / Feature	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.53	0.49	0.51	0.55	0.56	0.56	1253	1128	1309	2339
Token POS	0.53	0.52	0.53	0.57	0.57	0.57	1344	1188	1218	2329
Bigrams	0.53	0.54	0.54	0.58	0.59	0.58	1390	1233	1172	2368
Bigrams POS	0.55	0.46	0.50	0.59	0.57	0.57	1189	970	1373	2437
Trigrams	0.53	0.51	0.52	0.56	0.56	0.56	1296	1150	1266	2300
Trigrams POS	0.53	0.48	0.51	0.57	0.57	0.57	1240	1091	1322	2380
Dependency graph related										
Token-cue distance	0.66	0.67	0.66	0.74	0.73	0.73	1704	875	858	2942
First & second head POS	0.53	0.48	0.51	0.57	0.56	0.56	1235	1090	1327	2359
Ancestors (token, cue)	0.57	0.47	0.51	0.58	0.57	0.57	1200	909	1362	2456
Dependency path	0.70	0.52	0.60	0.71	0.71	0.70	1344	572	1218	3180

Conversely to what occurred in the case of aggregation, disaggregated lexical features actually perform better against the baseline than dependency-related features, with bigrams at 0.54 for in-scope F1 (baseline at 0.51), followed by token POS with values at 0.53, and trigrams with 0.52; while the dependency-related features first and second head POS and ancestor token-cue-token values stay at the baseline level of 0.51. However, token-cue distance and dependency path again outperform other features, raising the baseline for δ = 0.15 (from 0.51 to 0.66) for in-scope F1, and for δ = 0.17 (from 0.56 to 0.73) for all-labels average F1 in the case of token-cue distance; while the increase is of δ = 0.09 for in-scope F1 (at 0.60, baseline 0.51) and δ = 0.14 (with 0.70, baseline 0.56) for dependency path. Regarding confusion matrix values, individual in-scope tokens are better predicted using the token-cue distance, with the highest true positives at 1,704 (baseline of 1,253) and the lowest false positives (875) and false negatives (858).

- ***"Seldom"***

Table 33.A: Scope detection: Aggregated features performance for "seldom":

Measure Feature	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.40	0.41	0.40	0.56	0.59	0.57	641	960	934	2729
Token POS	---	---	---	---	---	---	---	---	---	---
Bigrams	0.44	0.46	0.45	0.60	0.61	0.61	728	914	847	2772
Bigrams POS	---	---	---	---	---	---	---	---	---	---
Trigrams	0.47	0.48	0.47	0.61	0.61	0.61	750	860	825	2777
Trigrams POS	0.47	0.49	0.48	0.62	0.62	0.62	766	866	809	2768
Dependency graph related										
Token-cue distance	0.64	0.64	0.64	0.79	0.79	0.79	1014	559	561	3510
First & second head POS	0.65	0.65	0.65	0.80	0.79	0.79	1025	543	550	3523
Ancestors (token, cue)	0.66	0.65	0.65	0.80	0.80	0.80	1028	536	547	3546
Dependency path	0.70	0.69	0.70	0.82	0.83	0.82	1090	471	485	3651

From the lexical features, bigrams and trigrams modestly raise the F1 scores from 0.40 for the in-scope baseline to 0.45 for bigrams (δ = 0.05) and 0.47 for trigrams (δ = 0.07), while the all-levels average values go from 0.57 at the baseline to 0.61 for both features (δ = 0.04). Confusion matrix scores also increase but moderately, by less than 100 values in each category. As the trend shows with the rest of the sub-corpora, values burst up when adding the dependency-related features, starting with an increase of δ = 0.24 (from 0.40 to 0.64) when adding the first dependency-related feature, token-cue distance, until reaching a δ = 0.30 raise (to 0.70) for in-scope F1; the all-labels average F1 also goes up, from 0.57 at the baseline to 0.79 for token-cue-distance (δ =

0.22), with a final score of 0.82 after adding dependency path (final δ = 0.25). Confusion matrix values also show a significant improvement, from 766 in true positives after adding the last lexical feature to 1,014 with token-cue distance and ending up with 1,090 after all dependency features are in place. False positives and false negatives lower substantially, from 866 and 809 for lexical features respectively, to 559 and 561 when adding the first dependency-related one and ending in 471 and 485 for each category.

Table 33.B: Scope detection: Individual feature performance for “seldom”:

Measure Feature	In-scope label values			Average values (all labels)			Confusion matrix for in-scope labels			
	Prec	Rec	F1	Avg. prec	Avg. rec	Avg. F1	True pos	False pos	False neg	True neg
Lexical										
Baseline: Token name	0.40	0.41	0.40	0.56	0.59	0.57	641	960	934	2729
Token POS	0.39	0.39	0.39	0.55	0.57	0.56	607	958	968	2729
Bigrams	0.44	0.46	0.45	0.60	0.61	0.61	728	914	847	2685
Bigrams POS	0.40	0.40	0.40	0.56	0.59	0.57	624	926	951	2772
Trigrams	0.46	0.44	0.45	0.60	0.61	0.60	700	829	875	2741
Trigrams POS	0.41	0.39	0.40	0.57	0.59	0.58	608	888	967	2788
Dependency graph related										
Token-cue distance	0.62	0.55	0.58	0.77	0.77	0.77	860	538	715	2776
First & second head POS	0.41	0.38	0.40	0.57	0.59	0.58	606	866	969	3552
Ancestors (token, cue)	0.45	0.45	0.45	0.60	0.64	0.61	703	865	872	2793
Dependency path	0.56	0.67	0.61	0.74	0.75	0.74	1054	813	521	2964

For this last corpus, the same trend of dependency-related features outperforming lexical ones still applies: regarding in-scope F1 values, token-cue distance shows a score of 0.58 while token-cue-path reaches 0.61 (δ = 0.18 and δ = 0.21 respectively); while values for all-labels average F1 are 0.77 for token-cue distance (δ = 20) and 0.74 for dependency path (δ = 17). Regarding lexical features, token POS actually lowers the F1 baseline values for both in-scope and all-labels average, with

0.39 and 0.56 (baselines of 0.40 and 0.57) respectively; for this feature, true positive values also decrease from 641 at the baseline to 607 which we suspect is because it is not an optimal feature for modeling negation prediction. Finally, “seldom” is the only sub-corpus in which the first and second head POS feature shows a significant improvement in the true negative values, being the highest with 3,552, followed by 2,793 for ancestor token-cue-token as second highest (baseline 2729). This may have happened due to the more complex layout of this approximate negator’s constituent structure in utterances, showing coordinated clauses and sub-clauses with several token heads as dependents of other token heads. In fact, tweets in the “seldom” corpus actually use more formal language than in the rest of the corpora, with plenty of literary quotations such as the following:

- “A proud man is seldom a grateful man, for he never thinks he gets as much as he deserves. Henry Ward Beecher” (SE950)

Interestingly, the improvement in true negatives seems to indicate a more effective performance in predicting true positives for cue and out-of-scope tokens; however, the true positives for in-scope actually decrease while the F1 for in-scope remains at the baseline.

Overall findings for automatic detection of scope negation. Overall, lexical features are less optimal for modeling in-scope negation prediction than dependency-related ones in the approximate negation corpus. Trigrams and bigrams show slight improvements in scores but are not sufficient to increase F1 values to satisfactory levels of above 0.60. On the other hand, dependency-related features across all approximate negator experiments outperform lexical ones, by remarkable delta values both from baseline (in disaggregated experiments) and from the last lexical feature added in accumulated values. Particularly token-cue distance and dependency path show the best performance improvement in all scores evaluation metrics and the lowest error values in the confusion matrices. The following table shows the delta improvement from the baseline values of these two best-performing features. Values are extracted from Table B, individual feature performance, under each approximate negator section. The delta value for the best performing lexical feature is also offered, for comparative purposes:

Feature	Improvement rates (δ)					
	Best dependency-related				Best lexical	
	Token-cue distance		Dependency path			
Sub-corpus	In-scope	All-labels avg	In-scope	All-labels avg	In-scope	All-labels avg
Barely	41	32	31	29	4 (trigrams)	4 (token POS)
Hardly	31	25	18	19	4 (trigrams)	4 (bigram POS)
Rarely	25	23	22	20	1 (trigrams)	--
Scarcely	15	17	9	14	4 (trigrams)	2 (bigram & trigram)
Seldom	18	20	21	17	3 (bigram)	2 (bigram)

Table 34: Improvement rates (δ) comparison between values for best dependency-related vs. best lexical features

We can see that the improvement rates for these dependency-related features is remarkable compared to lexical ones. Lexical features have a higher δ = 4 in most corpora, while the lowest delta differential for dependency feature 9, but then jump to 17, and with different values up to 71.

Summary of aggregated values. The following table summarizes the aggregated performance of all features, also incorporating information on cue and out-of-scope prediction scores as well as accuracy:

Measure / Approx. Negator	ICO[1] label values			Average values (all labels)			Confusion matrix			
	Prec	Rec	F1	Avg. prec	Avg. recall	Avg. F1	True pos[2]	Error values[3]	Total tokens processed (Support)	Accuracy[4]
Barely										
I	0.78	0.77	0.78	0.88	0.88	0.88	3843	540	(4383)	87.68%
C	0.81	0.91	0.86							
O	0.93	0.91	0.92							
Hardly										
I	0.76	0.71	0.74	0.86	0.86	0.86	9626	1540	(11166)	86.21%
C	0.74	0.87	0.80							
O	0.92	0.92	0.92							
Rarely										
I	0.67	0.62	0.65	0.82	0.82	0.82	4695	1029	(5724)	82.02%
C	0.64	0.67	0.65							
O	0.91	0.92	0.92							
Scarcely										
I	0.76	0.69	0.72	0.79	0.79	0.79	5023	1365	(6388)	78.63%
C	0.89	0.86	0.87							
O	0.79	0.85	0.82							
Seldom										
I	0.70	0.69	0.70	0.82	0.83	0.82	4741	1001	(5742)	82.57%
C	0.67	0.64	0.65							
O	0.91	0.92	0.91							

Table 35: Summary of aggregated scores for automatic detection of the scope of negation

Notes:

[1] I = in-scope tokens; C = negation cue; O = out-of-scope tokens

[2] True positives: values in the diagonal line of the confusion matrix

[3] Error values: false positives and false negatives for all labels

[4] Accuracy: estimated as the percentage of true positives from the total processed tweets

We can see that, in comparison, the classifier reaches the best performance in the "barely" sub-corpus, with higher precision, recall and F1 scores for both ICO (in-scope, cue, and out-of-scope disaggregated labels), and all-labels evaluation measures. It is worth mentioning that, in the case of disaggregated ICO label values, the classifier actually performs better when predicting out-of-scope (O) and cue (C) values instead of in-scope (I) ones.

Machine learning vs. human annotation scores. As with the negation labeling prediction experiments, machine learning results will now be compared to the human ceiling, i.e. human annotation scores gathered during simultaneous annotation of negation scope while developing the gold standard. The following table shows the number of strict in-scope token matches (i.e. the number of tokens considered under scope by both annotators), along with human and machine learning accuracy and F1 values for each subset of tweets. In the “strict in-scope matches” column, the total number of processes tweets is offered between square brackets to help understand the importance of that number:

Measure Sub-corpus	**Human Annotation**		**Machine Learning**	
	Strict in-scope matches [& Total simultaneously annotated tweets]	**Accuracy**		F1
Barely	59 [61]	96.72%	87.68%	0.88
Hardly	133 [142]	93.66%	86.21%	0.86
Rarely	67 [74]	90.54%	82.02%	0.82
Scarcely	51 [54]	94.44%	78.63%	0.79
Seldom	80 [83]	96.39%	82.57%	0.82

Table 36: Comparison between machine learning and human annotation scores for automatic detection of negation scope, by sub-corpus

The same limitations that applied to the comparison between human ceiling and machine learning performance in the case of automatic detection of negation still pertain to this comparison. Furthermore, while in the case of negation prediction the human annotated sample size was consistent across sub-corpora, in the case of scope those figures vary because the number of negated tweets also varies within different corpora. Consequently, the percentage of tweets annotated by humans for each sub-corpus is as follows: 25% for “barely,” 23% for “hardly,” 26% for “rarely,” 17% for “scarcely,” and 28% for “seldom.” Thus, the comparisons among values corresponding to different corpora should be made carefully. Overall accuracy values should also be taken with limitations in mind, since this evaluation measure does not reflect the impact of error over performance. In spite of those caveats, we can observe that the values in the table correspond to best practices in the industry and are, therefore, encouraging for explaining this phenomenon.

Closing remarks: answering the scope of the negation prediction component of the second research-sub question. As explained above, the second research sub-question states the following:

> Research sub-question 2: How do we automatically detect approximate negators that behave as reversal valence shifters (or prototype negators) in tweets?

This automatic detection task involved two phases: (i) predicting that an approximate negator acts as prototype negator, and; (ii) defining the scope of its negation influence over neighboring tokens in the tweet. This section offered findings regarding item (ii). As a result, we can conclude that the best way to model scope negation is to use dependency parsing features, along with lexical trigram and bigram ones. However, a caveat to this answer is that, although these dependency-related features show optimal F1 scores for detecting tokens within negation scope (between 0.62 and 0.72), they actually perform better when detecting tokens that are out of the scope of negation, with F1 results between 0.79 and 0.92. Although this tendency is commonly seen in standard English corpora results, which are highly imbalanced towards out-of-scope tokens (since in standard sentences the number of tokens negated tend to be less than the rest of the tokens), for the case of this corpus tweets could be shorter than regular sentences; thus (presumably), they could have shown a more balanced number of tokens within each category. Further research is, indeed, required.

Final Discussion: Answering the Primary Research Question

The results of this study discussed a series of methods and research strategies designed to answer the following research question:

> Research question: How do approximate negators operationalize reversal shifting (i.e. prototype or full) negation in Twitter utterances (tweets) in the context of improving automatic detection of negation in natural language processing?

The answer to this question can be drawn as follows: in general, an approximate negator’s partial valence shifting reversal effect on other words turns into full reversal when combined with items carrying negation emphasis power, such as negatively-oriented polarity-sensitive items (from the “any” item family). Additionally, for specific approximate negators (such as “rarely” and “seldom”), the use of other emphasis words (like the adverb “very”) can also create full negation. These words strengthen the approximate negator’s negative import, generating then a full prototype negation effect over other tokens. Finally, in the particular case of “seldom,” the syntax pattern of reversing the subject and auxiliary verb of its constituent offers a second type of full reversal operationalization.

Once the prototype negation is modeled as indicated above, details on the effect of that negation cue to neighboring tokens can be predicted by employing lexical (such as backward and forward bigram and trigram) features, combined with dependency graph ones. Regarding the dependency features, the classifier offered the best prediction performance with the following information as input for each token in focus: (i) number of other tokens separating it from the cue; (ii) dependency path from token in-focus to the negation cue; (iii) whether or not cue and token in-focus shared the same head in the dependency graph.

In spite of having arrived at an answer to the stated research question, limitations (derived from research design choices, type of the data, and other factors) to the accuracy and generalization of this answer still apply, and future work still needs to be performed to address missing elements. These topics will be discussed in the following chapter.

Chapter 5: Limitations and Future Work

The present study has contributed new knowledge to the subject of the operationalization of negation using non-prototype negation cues in colloquial speech, particularly in tweets. This concluding section will discuss the limitations of the study as well as future work which, in some cases, could overcome those limitations.

Limitations

Natural Language Processing Tools for Social Media Text

One of the limitations of the present research relates to the lack of options for tools capable of processing social media text whose colloquial style differs syntactically and grammatically from standardized language. As a result, robust NLP toolkits such as Stanford CoreNLP deal poorly with these types of corpora, even when using tools specifically developed for ungrammatical data such as the caseless models[18] (described in Chapter 3). At the same time, NLP tools for social media data, such as Tweebo (part of the Carnegie Mellon's ARK TweetNLP toolkit) offer limited syntactic dependency information, thus restricting the ability to draw detailed dependency relations. In consequence, more robust tools with proven performance across studies will be extremely helpful. Additionally, the standardized use of these tools across corpora could also facilitate the development of comparative studies. Finally, more tools for handling the emerging issue of robot activity-related messages could help clean data corpora faster, more accurately, and in a standardized fashion.

[18] https://stanfordnlp.github.io/CoreNLP/caseless.html

Biases Related to Feature Combinations

During the aggregated experiments, some features were deactivated due to low performance without exploring different aggregation combinations that could have yielded better results. One example is the feature reversed subject-auxiliary, aggregated after “very_rarely;” since in some tweets these two features overlap (such as in the case of “Very rarely do I call out of work or make excuses for myself”); the decision to run the former first undoubtedly penalized the latter and skewed results toward “very_rarely.” This decision was made to save iterations and expedite the experiment phase, but the limitation in the accuracy of results remains valid.

Modeling One Classifier Only

As explained in Chapter 3, a SVM classifier was trained due to its reported effective performance with social media and born-digital text. Since the results were satisfactory (above 0.60), no further model was explored for the experiments. However, future experiments could involve training a sequential classifier (such as CRF) to improve results further. This new model could potentially help improve in-scope prediction effectiveness which, as analyzed in the previous chapter, though satisfactory it has been the lowest among all scope labels with 0.65 F1 for the sub-corpus “rarely.”

Chronological Sampling

As mentioned in the “Sampling procedure” section, the sampling methodology employed was not random but chronological, i.e. using Twitter’s Streaming API, tweets were collected in real time following the sequence in which they were published by users. A random sampling would have involved collecting tweets during different moments over a period of time and then randomizing them in the database employing a chosen standard procedure. Although convenient for being easy to collect, chronological sampling procedure limits the generalization of results to phenomena occurring over the specific time period of collection. Furthermore, due to the temporary nature of social media content, the content represented by data analyzed in this research report may not be representative of Twitter conversations beyond the specific time of collection. In spite of that, findings from the inquiry on how negation operationalizes (through the syntactic and semantic patterns found) is still applicable.

Future Work

Impact on Sentiment Analysis Using Twitter Data

Indeed, one of the most important contributions to this study consists of its potential impact on sentiment analysis of tweets. As explained in the literature review, previous research on sentiment analysis has identified that modeling negation helps improve the overall performance of automatic sentiment analysis. With that in mind, the negation cue and dependency path features that proved to be most effective in negation prediction for this study could be engineered as input for sentiment detection tools to test their impact on performance. As reviewed in the surveyed literature, commonly used negation modeling approaches (such as negated contexts) add a prefix or suffix (typically "_neg") to those tokens identified as negation cue and scope, as a way to indicate their reversed polarity status. This information is then fed to the classifier as input to make prediction decisions on sentiment. Using the findings from the present research report, researchers could expand their negation model to include the new operationalizations of negation discussed in Chapter 4. A classifier could be trained to predict whenever tweets with approximate negators actually hold prototype negative polarity, and then determine the scope of negation using basic dependency paths drawn from assumed syntactic information (as roughly reflected in part-of-speech tags. The final output could then be treated as a negated context and pre-processed (such as adding "_neg") for the sentiment classifier to make more accurate decisions. Admittedly, this step was considered too ambitious for the scope of the present research design, but undoubtedly it constitutes the mandatory next step for future research.

Nuances Among Approximate Negators' Negative Import

As explained in the "Introduction" chapter ("Importance of the study" section), although recent research has identified the need to move away from the standard reversing assumption for negation modeling, differentiating the negative import of distinctive negation cues in order to model them separately (Zhu et al., 2014), this research did not move in that direction. In other words, the specific negative import of each of one of the five approximate negators "barely," "hardly," "rarely," scarcely," and "seldom" was not further explored. However, analyzing nuances in the negative effect of each one of these words (according to context, negated themes, syntax, and other factors), could potentially help make distinguishing among them better and push negation modeling even further forward.

Study of Part-of-speech Tags and Dependency Path Patterns for Scope and Cue Detection

Research on part-of-speech tag (POS) combinations that are more frequently annotated as in-scope in the gold standard could be the basis for the creation of new features to further improve the performance of the model. For instance, as Lapponi (2012) points out, the delexicalized structure of the final trigram in the negation clause "he never gives up," is adverb/conjugated verb/propositional verb particle or "RB/VBZ/PRT" in Penn Treebank POS tags. However, this POS pattern also matches affirmative expressions such as "always sleeps in" and "sometimes slows down." Consequently, backwards and forwards trigrams as features can only help the classifier assign probabilities effectively if more complementary information is offered, and that is when Lapponi decided to add the dependency path from cue to each token under scope as a feature.

In the case of the present research, the closest types of features engineered using a similar analytical approach of POS patterns are the reversed subject-auxiliary and the reversed polarity tag features for negation prediction. In this, grammatical patterns (such as adverb-verb-pronoun-verb, which is the first combination in that feature) have been selected from the literature as signaling negation, and then confirmed for that function in tweets during human annotation. However, additional research could be performed using tweets labeled as "Full" in the gold standard to then develop features based on new or more precise grammatical patterns of negation without prototype negation cues. In fact, the 12 POS tag combinations for the reversed subject-auxiliary feature used for the experiments were selected after analyzing the syntactic structure of true positive tweets when experimenting with combinations of those POS tags which most improved the classifier's performance.

Along the same lines, research on specific dependency path patterns pertaining to negation scope could help the classifier assign probabilities more effectively for that task. Research on combinations of nodes and edges found around negated scope constituents, such as N ↑ & ↑ V ↓ R (discussed in Chapter 3), could support discovering specific patterns related to those graph paths for negated constituents, with the final goal of developing more features to support the model's task.

Exploration of Language Usage Themes that Emerged from Annotation

During the development of the gold standard, annotators identified a series of themes emerging from manifestations of negations in tweets. Those themes can be organized into three groups: (i) use of sarcasm, (ii) negation in formal language style, and (iii) exaggeration. Additionally, the count of negated tweets that also contained at-mentions (including at-replies) or tokens showing the prefix character: "@," adds information about specific Twitter accounts related to this particular use of negation. At-mentions and at-replies involve any tweet that contains the "@" symbol followed by a Twitter handle; these are generally used to call attention to or publicly reply to another Twitter user (Bruns, & Stieglitz, 2014b; see also discussion below). The following table shows figures under each category:

Approximate Negator:	Theme coded as…			At-mentions (*)
	Sarcasm	Formal Language	Exaggeration	
Barely	13	0	8	93
Hardly	70	0	18	266
Rarely	15	0	2	118
Scarcely	46	64	32	112
Seldom	12	3	3	137
Total	**156**	**67**	**63**	**726**

Table 37: Count of themes emerged from annotation and at-mentions
(*) the column "at-mentions" counts the number of tweets showing the character "@" regardless the number of times it appears within the tweet (i.e. tweets with multiple at-mentions are counted only once)

It is worth mentioning that the code label "formal language" groups the codes "religious language," "poetic language," "style case," and "quotations." Conversely, the labels "sarcasm" and "exaggeration" did not group other codes.

An investigation into the nature of the aforementioned phenomena is beyond the scope of the present research. However, the frequency of their appearance along with some conceptualizations discussed in the literature review in Chapter 2 that closely relate to those emergent themes (such as Tottie's conceptualization of rejections and denials in verbal speech), helped to develop an initial examination for exploratory purposes. Subsequently, the following discussion briefly explains those emergent themes and their potential explanation from conceptualization in the pragmatic linguistic literature. This discussion could represent the initial steps to conducting further research in order to properly shed light on these phenomena.

First theme: use of irony and sarcasm. The dictionary definition of sarcasm is "a mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual" (Sarcasm, 2017). It follows that sarcasm is a particular type of ironic statement that is more aggressive in tone and also targets an individual or a class with the intent to hurt them (Kreuz & Glucksberg, 1989). Indeed, the pragmatic linguistic theory analyzes irony and sarcasm as a group, as manifestations of negation phenomena.

Paul Grice (1975) formulated a Standard Pragmatic Model that also provided a framework to explain irony. According to this Model, conversational utterances are based on the “cooperative principle,” a tacit agreement on some common purpose or shared direction between speakers. The conversational sub-principles of quantity, quality, relation, and manner derive from this principle. According to Grice, ironic utterances violate the sub-principle of quality, which indicates that both speaker and hearer agree upon telling the truth, or at least upon not telling false statements.

Horn (1989) expands upon Grice’s cooperative principle to explain how negation works, including its relation to irony and sarcasm. He reduces Grice’s four sub-principles into two: “quantity” (which dictates that the speaker should provide the sufficient amount of information for the communication task to be effective), and “relation,” or the need to make the information contribution relevant to the present conversation. There is a constant dialectic and functional tension between these two sub-principles, and utterer and hearer constantly move from one to the other making inferences about what is said in the conversation. When inferences are governed by the quantitative sub-principle, enough information is offered to convey a specific message; thus, the hearer assumes that if a speaker does not use a more informative form of the utterance, it is because he or she was not epistemically equipped to do so. In terms of logic, “when a speaker is saying ‘…P_i…’ implicates that for all s/he knows ‘…at most

P_i…' that is, that is not the case that '…P_j …' for any P_j stronger than P_i" (Horn, 1989, p. 195). Hence, the quantitative sub-principle rules the communication in the sense that there is an economy of information. Conversely, relation-ruled inferences give more license to the hearer to add contextual elements from his or her background, shared information, etc.; in that sense, this style of communication follows Levinson's principle, which states that hearers "read as much into an utterance as is consistent with what you know about the world" (Levinson as quoted by Horn, 1989, p. 196). Under this sub-principle, then, when a speaker says '…P_i…' it actually implicates '…P_j …' in relation to "some P_j stronger than P_i and/or representing a salient subcase of P_i" (Horn, 2001, p. 195). In the particular case of ironic utterances, Horn explains, this translates into a speaker uttering content related to some unwanted or disagreeable situation (P_j) by making a vaguer and/or more digestible statement (P_i), and leaving to the hearer the task to strengthen the inference back to the more specific, intended content (P_j). This intended content is meant to be semantically more negative than the actual utterance, such as in the case of "a lady of a certain age" or "a woman who is no longer young" to describe an old woman (Horn, 1989, p. 338). In the case of sarcasm, such negative content exacerbates the sentiment and it will also identify a clear victim or target of that negativity.

Horn also states that an important element in this formulation of irony is that this emphasis on the hearer and his or her background causes relation-based utterances to contain sociocultural rather than linguistic motivations. As part their Pretense Theory of Irony, Clark and Gerrig (2007) discuss that this shared context and/or sociocultural background draws an inner-circle of complicity between utterer and hearer, leaving outside in an outer circle those hearers who do not share the same background.

This relation-based, socio-culturally motivated type of inference in pragmatic communication offers a framework that helps us understand tweets labeled as "sarcasm" in the corpus. Some examples from each approximate negator's sub-corpus will be analyzed to exemplify this phenomenon:

- "City of brotherly love my ass. Douchebags are barely a half step above Jersey." (BA1333)

In this case the utterer uses "barely" to state that the "Douchebags" of Philadelphia are only a little better than "Jersey" (statement P_i), but the insulting tone of the preceding sentence strengthens the negative import of "barely," making the hearer infer that the real meaning is that the former is worse than the latter (P_j). Thus, the final inferred meaning is more negative than the one explicitly conveyed by the utterer (as Horn explains). Notice that the hearer requires cultural background to make the right inference: "City of brotherly love[19]" is a popular name for the city of Philadelphia and "Douchebags" is a derogatory nickname, in this case, for Philadelphians[20]. A possible inference will rise from a conversation about sports as a context, particularly hockey, with speakers sharing their emotions about Philadelphia and New Jersey teams. Those hearers with sports backgrounds will make that inference and be part of Clark and Gerrig's inner circle of ironic communication, while those who do not share the same background are left to the outer circle, missing that inference.

[19] https://en.wikipedia.org/wiki/Philadelphia and also: http://philadelphiaencyclopedia.org/archive/city-of-brotherly-love/

[20] https://www.thrillist.com/entertainment/philadelphia/signs-you-re-a-philly-douchebag-worst-people-in-philly

The following tweet illustrates another relation-ruled inference found in the corpus:

- “@__simplysb 🖕■🖕■ nothing nigga im clean .... Barely😂” (BA1319)

In this case, the utterer indicates that s/he is “clean” (statement P_i) but adds the approximate negator at the end with a literal vague meaning that, added to the positive emoticon and the coarse language preceding, actually leads to the inference: “I am not clean” (P_j).

Use of “framing.” A set of tweets labeled as sarcasm showed what Clift (1999) defines as “framing,” which occurs when a speaker echoes somebody else’s words by placing them between quotation marks or some other type of punctuation. In that case, he or she separates and puts distance between his or her own opinion and the word’s literal content, typically as a way to show disapproval:

- "@KennedyNation @Curly_gurl135 @Detron3000 @dorimonson Amusing perhaps. Drunk Girls are seldom ‘amazing’, Except to themselves. Amusing tho." (SE1299)

By framing “amazing” between quotation marks, the utterer shows that the attribute for “drunk girls” is given to him or her, but s/he remains skeptical or plainly does not believe it and decided not to take those words as theirs but rather to quote them from somebody else’s statement.

- “My accent is hardly “strange”! Everyone speaks like this in my home country!” (HA332)

Framing the adjective “strange” helps the utterer show skepticism about being a suitable qualification for “accent.” The humorous second clause is a polite way of giving context to the first one, and helps the hearer arrive to the inference that the utterer's accent should not be taken as strange.

Second theme: dialogical negation. As shown above, 726 tweets from the 1,766 labeled as Full also showed one or several "at" ("@") prefixed words, (representing 41% of the negated tweets corpus). This character precedes a user name profile in the Twitter platform (such as "@norespal") and it is used with two main goals: (i) as "at-mention" to refer to that particular user in an utterance, with or without the intention to start a dialog, (ii), as "at-reply" to address a message directly to that specific twitterer, on a more dialogical fashion. Although in some cases it can be difficult to distinguish between these two usages (Bruns, & Stieglitz, 2014b), it is clear that this affordance allows a specific twitterer to make another Twitter user as part of a given utterance, either as a recipient of the message or the target of its content. The fact that there exists one or multiple addressees and/or targets of a message, along with the casual and dialogical nature of Twitter utterances, can situate this phenomenon among those explained by Tottie (1991) in her pragmatic study of negation in speech corpora (discussed in the literature review). It is worth mentioning that, due to its descriptive and quantitative approach along with the use of speech data (the London-Lund and Lancaster-Oslo/Bergen corpora), Tottie's linguistic framework was employed by natural language researchers such as Councill et al. (2010, July) to raise understanding of speech-related online content such as product reviews.

Tottie’s analysis of speech discourse versus written text indicates that there is twice as much negation in the former than in the later. To explain this fact, she proposes the following discourse-functional classification of negative sentences:

(i) Rejections (including refusals)
(ii) Denials:
  (a) Explicit
  (b) Implicit

(Tottie, 1991, p. 22)

As already discussed, rejections involve declining direct proposals (or not complying with a request, in the case of refusals), while denials negate assertions about situations. Denials can take two forms: (i) explicit, consisting of unambiguous assertions or situations; and (ii) implicit, where expectations are not met or contextual conditions make the utterer negate tacit elements. Notice again the emphasis on contextual factors in implicit denials, which resembles the socio-cultural component of Horn’s relation-ruled inferences, as well as Clark and Gerrig’s (2007) pretense theory of inner-circles in ironic communication, circles defined by a common shared background.

Tottie argues that, in the specific case of spoken English, rejections (including refusals) and explicit denials are more frequent than in written English. In the particular case of rejections, polite refusals help speakers cooperate in continuing the conversation (under Grice’s cooperative principle) while refusing a presented status, an offer, etc.

Application to the present research. The following tweets (taken from the gold standard) contain at-mentions and have also been labeled as “Full” (or negated tweets) by annotators. At the same time, exemplary tweets such as the first explicit denial offered below (HA374) do not show any of the reoccurrences that signal negation as described by this research. However, as we will see, Tottie's framework of dialogical negation could help draw an explanation for the negative import conveyed by the approximate negators, as follows:

Rejections:

- "@mookiealexander I don't buy into it either, but beating Hendo hardly proves anything to me." (HA1384)

The Twitter user here seems to indicate that, although he agrees with the receiver in the first part of the statement, s/he disagrees with some previously uttered content related to Hendo; s/he uses "hardly" and the NPI "anything" to convey a rejection statement.

Polite refusals:

- "@DebdebWilder @Archivist1000 @jimsciutto The Huffington Post is hardly Fox News, wouldn't you agree?" (HA1394)

The fact that the there is an invitation to agree in the final interrogatory constituent attached to the clause (in an elaborated form of reversed polarity tag) seems to indicate that "hardly" politely invites to negate that the subject (Huffington Post) has the same properties as the predicate (Fox News).

Explicit denials:

- "@DreTownes Racism was hardly started by white people. That's an incredibly ignorant thing to say." (HA374)

The second clause indicates that what is said in the first one proves a level of ignorance so high that it is incredible that anyone could be so ignorant. Such hyperbolic expression invalidates the credibility of what is said in that first clause, emphasizing the negative import of “hardly” and, thus, denying such content. Notice the absence of specific tokens that could act as negation cues. Indeed, the study of patterns in this style of explicit denial could support discovering new forms of negation.

- “@Nowdied2 OMG! You serious? No suprise hardly anyone follows you.. Piss taking Irony Joking Sexism All the above.” (HA1183)

The acronym OMG (colloquial acronym for “oh my God”) expressing shock or disbelief, along with the request for confirming what has been heard (“You serious?”), collaborate with the sarcastic tone of “hardly anyone” to indicate that actually no one follows this particular user, emphasizing thus the negative import of the approximate negator. The entire tweet aims to deny what @Nowdied2 has uttered to the receiver.

As mentioned at the beginning of this section, the topics of irony and sarcasm, as well as dialogical negation, exceed the focus of the present report and its research question. However, emergent themes in corpora, as identified and commented upon by annotators during the development of the gold standard corpus, made imperative the dedication of a brief section to discuss them. Their in-depth research and exploration should be the focus of future studies.

References

Balikas, G., & Amini, M. R. (2016). *TwiSE at SemEval-2016 Task 4: Twitter Sentiment Classification.* arXiv preprint arXiv:1606.04351

Balikas, G. (2017). TwiSe at SemEval-2017 Task 4: Five-point Twitter Sentiment Classification and Quantification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 755-759

Baziotis, C., Pelekis, N., & Doulkeridis, C. (2017). Datastories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 747-754

Blanco, E., & Moldovan, D. I. (2011, March). Some Issues on Detecting Negation from Text. In *FLAIRS Conference.* Retrieved from: http://www.cse.unt.edu/~blanco/papers/issues_detecting_negation.pdf

Boag, W., Potash, P., & Rumshisky, A. (2015). Twitter-Hawk: A feature bucket approach to sentiment analysis. In *SemEval@ NAACL-HLT*, 640-646.

Bruns, A. & Stieglitz, S. (2014a). Twitter data: What do they represent?. *Special Issue: Social Media / Katrin Weller, Markus Strohmaier. it - Information Technology, 56*(5), 240-245. Available at doi:10.1515/itit-2014-1049

Bruns, A., & Stieglitz, S. (2014b). Metrics for understanding communication on Twitter. In *Twitter and Society.* New York: Peter Lang, Chapter 6, 69-82

Chakraborty, M., Pal, S., Pramanik, R., & Chowdary, C. R. (2016). Recent developments in social spam detection and combating techniques: A survey. In *Information Processing & Management, 52*(6), 1053-1073

Chowdhury, M., & Mahbub, F. (2012, June). FBK: Exploiting phrasal and contextual clues for negation scope detection. In Association for Computational Linguistics. (2012). *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, 340-346

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg?. In *IEEE Transactions on Dependable and Secure Computing,* 9(6), 811-824

Clark, H. H. & Gerrig, R. J. (2007). On the pretense theory of irony. In R. W. Gibbs Jr. & H.L. Colston (Eds.), *Irony in language and thought: A cognitive science reader,* 25-33. New York, NY: Lawrence Erlbaum Associates

Cliche, M. (2017). *BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs*. Available at arXiv preprint arXiv:1704.06125

Clift, R. (1999). Irony in conversation. *Language in society, 28*(04), 523-553

Councill, I. G., McDonald, R., & Velikovich, L. (2010, July). What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In Association for Computational Linguistics. (2010). *Proceedings of the workshop on negation and speculation in natural language processing*, 51-59

Creswell, J. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches*. (4th Ed.). Thousand Oaks, CA: Sage

Cruz, N. P., Taboada, M., & Mitkov, R. (2016). A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology, 67*(9), 2118-2136

De Marneffe, M. C., & Manning, C. D. (2008). *Stanford typed dependencies manual*: *Revised for the Stanford Parser v. 3.7.0 in September 2016.* Technical report, Stanford University. Retrieved from: https://nlp.stanford.edu/software/dependencies_manual.pdf

Da San Martino, G., Gao, W., & Sebastiani, F. (2016). QCRI at SemEval-2016 Task 4: Probabilistic Methods for Binary and Ordinal Quantification. In *SemEval@ NAACL-HLT*, 58-63

Das, S., & Chen, M. (2001, July). Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the 8th Asia Pacific finance association annual conference, 35.* Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.202.6418&rep=rep1&type=pdf

De Swart, H. (2010). *Expression and Interpretation of Negation: An OT Typology.* London: Springer

Driscoll, K., & Walker, S. (2014). Big Data, Big Questions| Working Within a Black Box: Transparency in the Collection and Production of Big Twitter Data. *International Journal of Communication, 8* (20), 1745-1764

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM, 59*(7), 96-104

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J. & Smith, N. A. (2011, June). Part-of-speech tagging for twitter: Annotation, features, and experiments. In Association for Computational Linguistics. (2011)*. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, 42-47

Gimpel, K., Schneider, N., & O'Connor, B. (2013). *Annotation Guidelines for Twitter Part-of-Speech Tagging, Version 0.3 (March 2013)*. Available at the Tweet NLP-ARK-Carnegie Mellon website: http://www.cs.cmu.edu/~ark/TweetNLP/annot_guidelines.pdf

Grice, H. P. (1975). Logic and conversation. In P. Cole (Ed.), *Syntax and semantics 3: Speech acts*, 41-58. New York: Academic Press.

Haegeman, L. (1995). *The Syntax of Negation*. (Cambridge Studies in Linguistics; 75).

Hamdan, H., Bellot, P., & Bechet, F. (2015, June). Lsislif: CRF and Logistic Regression for Opinion Target Extraction and Sentiment Polarity Analysis. In *SemEval@ NAACL-HLT*, 753-758

Harabagiu, S., Hicki, A. & Lacatusu, F. (2006, July). Negation, contrast and contradiction in text processing. In *AAAI, 6*, 755-762

Hemsley, J., Ceskavich, B., Tanupabrungsun, S. (2014). *STACK (Version 1.0).* Syracuse University, School of Information Studies. Retrieved from https://github.com/bitslabsyr/stack DOI: 10.5281/zenodo.12388

Horn, L. (1989). *A natural history of negation.* Chicago, IL: CSLI Publications

Horn, L. & Kato, Y. (2000). Introduction: Negation and Polarity at the Millennium. In Horn, L. & Kato, Y. (Eds.). (2000). *Negation and Polarity: Syntactic and Semantic Perspectives.* New York, NY: Oxford University Press.

Horn, E. (2011). Multiple negation in English and other languages. In Horn, L. (Ed.). *The expression of negation.* (The Expression of Cognitive Categories Series; ECC 4). New York, NY: De Gruyer Mouton

Huddleston, R. (2002). Clause type and illocutionary force. In *The Cambridge Grammar of the English Language,* Chapter 10, 851-945

Jespersen, O. (1917). *Negation in English and other languages.* Copenhagen: A. F. Host

Jia, L., Yu, C., & Meng, W. (2009, November). The effect of negation on sentiment analysis and retrieval effectiveness. In *ACM, Proceedings of the 18th ACM conference on Information and knowledge management*, 1827-1830

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall series in artificial intelligence

Kennedy, A., & Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, *22*(2), 110-125

Kiritchenko, S., Zhu, X., Cherry, C., & Mohammad, S. (2014, August). NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In Association for Computational Linguistics. (Dublin, Ireland, August 23-24, 2014). *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014),* 437–442

Klima, E. (1964). Negation in English. In Fodor, J. & Katz, J. (Eds) (1964). *The structure of language: Readings in philosophy of language.* Englewood Cliffs, NJ: Prentice-Hall, 246-323

Kolovou, A., Kokkinos, F., Fergadis, A., Papalampidi, P., Iosif, E., Malandrakis, N., Palogiannidi, E., Papageorgiou, H., Narayanan, S. & Potamianos, A. (2017). Tweester at SemEval-2017 Task 4: Fusion of Semantic-Affective and pairwise classification models for sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 675-682

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., & Smith, N. A. (2014). A dependency parser for tweets. In Association for Computational Linguistics. (2014). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (October 25-29, 2014, Doha, Qatar), 1001–1012

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

Krippendorff, K. (2013). *Content Analysis: An Introduction to Its Methodology.* Thousand Oaks, CA: Sage

Lango, M., Brzezinski, D., & Stefanowski, J. (2016). PUT at SemEval-2016 Task 4: The ABC of Twitter Sentiment Analysis. In *SemEval@ NAACL-HLT*, 126-132

Lapponi, E. (2012). *Why Not!: Sequence Labeling the Scope of Negation Using Dependency Features.* [Master's Thesis]. University of Oslo, Department of Informatics

Lapponi, E., Velldal, E., Øvrelid, L., & Read, J. (2012, June). Uio 2: sequence-labeling negation using dependency features. In Association for Computational

Linguistics, *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation,* 319-327

Lapponi, E., Read, J., & Øvrelid, L. (2012, December). Representing and resolving negation for sentiment analysis. In *2012 IEEE 12th International Conference, Data Mining Workshops (ICDMW),* 687-692

Lavergne, T., Cappé, O., & Yvon, F. (2010, July). Practical very large scale CRFs. In Association for Computational Linguistics. (2010). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* 504-513

Leedy, P. D and Ormrod, J.E. (2013). *Practical Research: Planning and Design*. (10th Edition). New Jersey: Pearson

Li, Q., Nourbakhsh, A., Liu, X., Fang, R., & Shah, S. (2017). funSentiment at SemEval-2017 Task 4: Topic-Based Message Sentiment Classification by Exploiting Word Embeddings, Text Features and Target Contexts. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 741-746

Liu, B. (2011). *Web data mining: Exploring hyperlinks, contents and usage data.* Chicago, IL: Springer

Liu, J., & Seneff, S. (2009). Review sentiment scoring via a parse-and-paraphrase paradigm. In *Association for Computational Linguistics, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, (1),* 161-169

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press

Manning, Ch. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60. Retrieved from: https://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf

McCracken, N. (2015). Basic text processing: Sentence segmentation (power point slides). In *Natural Language Processing.* (Online course). Syracuse, NY: iSchool, University of Syracuse

Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013, June). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SemEval '13).* Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.310.7022&rep=rep1&type=pdf

Morante, R., & Blanco, E. (2012). * SEM 2012 shared task: Resolving the scope and focus of negation. In Association for Computational Linguistics. (2012). *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, 265-274

Morante, R., Schrauwen, S. and Daelemans, W. (2011, May). Annotation of negation cue and their scope: Guidelines v1.0. *Computational Linguistics and Psycholinguistics Technical Report Series, CLIPS (CTRS-003)*

Morante, R and Sporleder, C. (2012). Modality and negation: An introduction to the special issue. *Computational Linguistics, 38*(2), 223-260

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016, June). SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *SemEval@ NAACL-HLT*, 1-18

Nivre, J. (2010). *Inductive Dependency Parsing.* Dordrecht: Springer. (Text, Speech and Language Technology; v. 34)

Nivre, J & Kubler, S. (2006). *Dependency Parsing: Tutorial at COLING-ACL, Sydney 2006.* Available at: https://stp.lingfil.uu.se/~nivre/docs/ACLslides.pdf

Neuendorf, K. (2002). *The Content Analysis Guidebook.* Thousand Oaks, CA: Sage

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., & Smith, N. A. (2013) *Improved part-of-speech tagging for online conversational text with word clusters*. Association for Computational Linguistics

Palogiannidi, E., Kolovou, A., Christopoulou, F., Kokkinos, F., Iosif, E., Malandrakis, N., Papageorgiou, H., Narayanan, S., & Potamianos, A. (2016). Tweester at SemEval-2016 Task 4: Sentiment Analysis in Twitter Using Semantic-Affective Model Adaptation. In *SemEval@ NAACL-HLT*, 155-163

Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 79-86. Association for Computational Linguistics

Payne, J. and Huddleston, R. (2002). Nouns and noun phrases. In: *The Cambridge Grammar of the English Language*, Chapter 5, 323-523

Plotnikova, N., Kohl, M., Volkert, K., Evert, S., Lerner, A., Dykes, N., & Ermer, H. (2015, June). KLUEless: Polarity Classification and Association. In *SemEval@ NAACL-HLT*. 619-625

Polanyi, L. and Zaenen, A. (2004). Contextual valence shifters. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text*, 106-111

Potts, Ch. (2011, November 8-9). “Sentiment Symposium Tutorial,” in *Sentiment Analysis Symposium*, San Francisco, CA: Stanford University. Available at: http://sentiment.christopherpotts.net/index.html

Pullum, G. and Huddleston, R. (2002). Negation. In: *The Cambridge Grammar of the English Language*, Chapter 5, 785-851

Pustejovsky, J., & Stubbs, A. (2012). Natural language annotation for machine learning. Sebastopol, CA: O'Reilly Media, Inc.

Ramshaw, L. A., & Marcus, M. P. (1995). Text chunking using transformation-based learning. In: *Proceedings of the Third ACL Workshop on Very Large Corpora*, ACL. Retrieved October 12, 2015, from: ftp://ftp.cis.upenn.edu/pub/chunker/wvlcbook.ps.gz

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., & Stoyanov, V. (2015, June). SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *SemEval@ NAACL-HLT*, 451-463

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

Saldana, J. (2013). *The coding manual for qualitative researchers*. Sage.

Sarcasm. (2017). In *Merriam-Webster.com*. Retrieved November 24, 2017, from https://www.merriam-webster.com/dictionary/sarcasm

SemEval-2015, Task 10: Sentiment Analysis in Twitter. (n.d.). Retrieved from: http://alt.qcri.org/semeval2015/task10/

SemEval-2016, Task 4: Sentiment Analysis in Twitter. (n.d.). Retrieved from: http://alt.qcri.org/semeval2016/task4/

SemEval-2017, Task 4. Sentiment Analysis in Twitter. (n.d.). Retrieved from: http://alt.qcri.org/semeval2017/task4/

SemEval-2018: International Workshop on Semantic Evaluation. (2018). *Tasks.* Retrieved from: http://alt.qcri.org/semeval2018/index.php?id=tasks

Severyn, A., & Moschitti, A. (2015, June). UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In *SemEval@ NAACL-HLT*, 464-469

Shaikh, M. A. M., Prendinger, H., & Mitsuru, I. (2007). Assessing sentiment of text by semantic dependency and contextual valence analysis. In *Affective Computing and Intelligent Interaction*, 191-202

Stanford CoreNLP (n.d.). *Core NLP (version 3.8.0): Caseless models*. Retrieved April 16, 2017 from the Stanford Natural Core NLP web site, available at: https://stanfordnlp.github.io/CoreNLP/caseless.html

Stanford CoreNLP (n.d.). *Stanford CoreNLP – Natural language software: Human languages supported*. Available at: https://stanfordnlp.github.io/CoreNLP/index.html

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics, 37*(2), 267-307.

Tottie, G. (1991). *Negation in English speech and writing: A study in variation.* Quantitative Analyses of Linguistic Structure Series. San Diego, CA: Academic Press

Tracy, S. J. (2013). *Qualitative research methods: Collecting evidence, crafting analysis, communicating impact*. Malden, MA: John Wiley & Sons

Twitter. (2017). Twitter Developer Documentation: Tweets. Available at: https://dev.twitter.com/overview/api/tweets

Twitter Developers Forum. (2014, February). *How to get the user's device information of a tweet via Streaming API?.* Available at: https://twittercommunity.com/t/how-to-get-the-users-device-information-of-a-tweet-via-streaming-api/15012

Van der Auwera, J. (2011). On the diachrony of negation. In: Horn, L. (Ed.). *The Expression of Negation*. Berlin: Walter de Gruyer (The expression of cognitive categories; 4)

White, J. P. (2012, June). UWashington: Negation resolution using machine learning methods. In Association for Computational Linguistics. (2012). *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation,* 335-339

Wiegand, M., Balahur, A., Roth, B., Klakow, D., & Montoya, A. (2010). A survey on the role of negation in sentiment analysis. In Association for Computational

Linguistics. *Proceedings of the workshop on negation and speculation in natural language processing*, 60-68

Xu, S., Liang, H., & Baldwin, T. (2016). UNIMELB at SemEval-2016 Tasks 4A and 4B: An Ensemble of Neural Networks and a Word2Vec Based Model for Sentiment Classification. In *SemEval@ NAACL-HLT*, 183-189

Yang, C., Harkreader, R. C., & Gu, G. (2011, September). Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In *International Workshop on Recent Advances in Intrusion Detection (RAID),* Springer Berlin Heidelberg, 318-337

Zhou, Y., Zhang, Z., & Lan, M. (2016). ECNU at SemEval-2016 Task 4: An Empirical Investigation of Traditional NLP Features and Word Embedding Features for Sentence-level and Topic-level Sentiment Analysis in Twitter. In *SemEval@ NAACL-HLT*, 256-261

Zhu, X., Guo, H., Mohammad, S., & Kiritchenko, S. (2014). An empirical study on the effect of negation words on sentiment. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (14)

Zhu, X., Kiritchenko, S., & Mohammad, S. M. (2014, August). NRC-Canada-2014: Recent improvements in the sentiment analysis of tweets. In Association for Computational Linguistics. (Dublin, Ireland, August 23-24, 2014). *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014),* 443–447

Norma Estela Palomino

| Innovation Catalyst
| Project Team Synergist
| Computing Linguistics Researcher

VITA

norma.palomino@gmail.com
nepalomi@syr.edu

Education

- Doctor of Professional Studies in Information Science (DPS). May 2018
  - *2018 Doctoral Prize Recipient, Information Science and Technology Program*
- Master of Science in Information Studies (MSIS), University of Texas at Austin, 2003
- B.A. in Philosophy, University of Morón (Argentina), 1993

HIGHLIGHTED TEACHING AND SCHOLARLY EXPERIENCE

Lecturer

- Designed the syllabus and didactic approach for "Datología: data for public policy making effectiveness" [Spanish], an edX-IDB MOOC on foundations of statistical methods and data literacy for evidence-based decision-making processes for Spanish speaking public policy makers. Available at: https://www.edx.org/course/datos-para-la-efectividad-de-las-idbx-idb10x
- Contributed in the development of the Scripting Application Workshop (Prof. Nancy McCracken), May 2015, Information School, Syracuse University;
- Advanced Course in Library Management (parts 1 and 2): Strategic Planning for Libraries [in Spanish], Graduate School of Library Science, National University of La Plata, Argentina, Fall & Spring Semesters, 2006

Conference Presentations

- Poster: Digital Scholarly Cycle Disruptions and the Academic Library: Challenges and Opportunities. (2017, October) [Discussing the phenomenon of fake news in social media in the context of information literacy]. Frankfurt, Germany: New Directions for Libraries, Scholars, and Partnerships: an International Symposium. Center for Research Libraries (CRL). Program available at: https://www.crl.edu/program
- Keynote: Leadership and the Library Profession. (2017, September). Rio de Janeiro, Brazil: Fórum de Inovação e Empreendedorismo na Biblioteconomia (FIEB): Tema-Atitude Empreendedora [Brazilian Association of Librarian Entrepreneurs Forum]. Program available at: http://www.fieb.net.br/
- "Open Data at the IDB" [Spanish], presenter, V Library and Information Sciences Conference (Lima, Peru), November, 2014

Publications

- "How to use data analytics to identify high school student performance issues" [Spanish] [Blog Post]. Blog "Abierto al Público", IDB, October 5, 2017. Available at: https://blogs.iadb.org/abierto-al-publico/2017/10/05/analisis-de-datos-mooc/;
- Co-author: "Using Big Data and its Analytical Techniques for Public Policy Design and Implementation in Latin America and the Caribbean" [Technical Note]. Washington, DC: Inter-American Development Bank (IDB). Available at: https://publications.iadb.org/handle/11319/8276;

- Co-author: "Creating an Open Data Portal", [Magazine Article]. Information Outlook, January-February 2015 issue;
- "5 ideas about and beyond Open Access Week" [Blog Post]. Blog "Abierto al Público", IDB, October 23, 2015. Available at: http://blogs.iadb.org/abierto-al-publico/2014/10/23/5-ideas-acerca-y-mas-alla-del-acceso-abierto/;
- "The IDB will launch its open data portal on development in Latin America and the Caribbean" [Blog Post]. Blog "Abierto al Público", IDB. December 11, 2014. Available at: http://blogs.iadb.org/abierto-al-publico/2014/12/11/idb-will-launch-open-data-portal-development-latin-america-caribbean/;
- "10 ways to improve lives using big data" [Spanish] [Newspaper Article]. "El País: Planeta Futuro" [Spanish mainstream newspaper]. April 25th 2014. Available at: http://elpais.com/elpais/2014/04/25/planeta_futuro/1398424819_252681.html;
- "Latin America at the forefront of the open access movement" [Spanish] [Blog Post]. Blog "Abierto al Público", Knowledge and Learning Department, IDB, March 4, 2015. Available at: http://blogs.iadb.org/abierto-al-publico/2014/03/04/america-latina-en-la-vanguardia-del-acceso-abierto/;
- Co-author: "Righting the academic paper: a collaboration between library services and the writing centre in a Canadian academic setting" [Peer Reviewed Journal Article]. New Library World, *112*(3/4), 2011; and
- "Information for action: the approach used by NGOs to foster and strengthen civic life: A case study from Argentina" [Book Chapter]. "Changing Roles of NGO's in the Creation, Storage, and Dissemination of Information in Developing Countries." Steve W. Witt (Ed). Zürich: K.G. Saur, 2006 (IFLA publications; 123).

HIGHLIGHTED EXECUTIVE EXPERIENCE

- 25+ years of international experience effectively providing vision and executive direction in the development of innovative knowledge-oriented products and services
- Research experience in computing linguistics using natural language processing tools and supervised and unsupervised machine learning to predict patterns in corpora. Experience applying predictive analytics models to social media data
- Extensive experience handling knowledge-based digital assets, including database management, information interoperability, metadata handling, and the like. Solid experience in the implementation of key performance indicators along with visually effective platforms for communication and tracking
- Successfully led 5 institutional modernization projects, re-engineering processes, updating technologies, and managing change towards catalyzing innovation
- International experience leading multicultural teams and establishing collaboration among institutional partners to realize corporate goals
- Effectively re-engineered unit's budgeting according to new institutional guidelines on results-based budgeting, accountability, efficiency and effectiveness

Professional Accomplishments

Unit Chief, Information Services Unit, Inter-American Development Bank (IDB), Washington, DC., 2010- 2018

✓ Established the foundational structure of the Unit involving three core business areas: knowledge platforms, data analytics, and library services

✓ Supervised 25 team members comprising data scientists, visualization experts, data analysts, programmers, web designers, semantic web practitioners, information scientists, and librarians, in charge of the institutional website, knowledge and data repositories, key performance indicator dashboards, and research report workflow applications, supported by an overall budget of $4 million

✓ Oversaw negotiations and management of service level agreements, requests for proposals, and maintenance contracts with 7 vendors to support a variety of platforms and technologies

✓ Led the development of a comprehensive knowledge metrics portfolio to track KPI related to the impact of knowledge products dissemination

✓ Oversaw the customization of social media analytics tools (Brandwatch) for tracking digital dialogs pertinent to institutional strategies (such as following specific topics in Spanish or Portuguese)

✓ In collaboration with a variety of institutional partners, envisioned, designed, and executed the first IDB open data platform featuring more than 1,700 socio-economic indicators and 90+ datasets. The portal includes innovative animated and interactive visualizations as well as a comprehensive set of data exchange tools

✓ Led the revamp of the Institutional Knowledge Repository (BRIK) and Publications website, incorporating cutting-edge features such as Linked Data for dissemination and text mining capabilities for automatic subject labeling of documents

✓ Led the development of a machine learning model that uses supervised and unsupervised learning for classifying institutional documents under a records management scheme. The solution's pilot effectively classified 2,000 documents (the final solution will be applied to 1.5 million digital objects)

✓ Led the development of knowledge product workflows in the organization's mainstream processes application (called "Convergence"). Acted as business leader advising in decision-making processes regarding data and metadata transfer, technical interoperability, document management, and the like

✓ Implemented search engine optimization (SEO) techniques knowledge product findability on the web, which increased the visibility of Bank's publications in Google Scholar by 74%

✓ Led a 3-year, $ 2.5 million capital project to create a knowledge recommendation system based on user's digital traces, using semantic web and machine learning technologies embedded at the back end of multiple corporate platforms such as the institutional document repository, institutional document management system, and external website

✓ Re-engineered the Unit's budget according to new institutional guidelines on results-based budgeting, accountability, efficiency and effectiveness

✓ Led the update of information management technologies by implementing a discovery layer product (Ex-Libri's PRIMO) and revamping the new library website, conducting usability tests and building "persona" profiles

Manager of Library Services, University of Guelph-Humber (UofGH, *A joint venture between the University of Guelph and Humber College*) Toronto, Ontario, Canada, 2008-2010

✓ Led the UofGH strategic planning process to update the vision, mission, values and goals for information services and programs. Developed the final Strategic Plan document and delivered presentations to engage stakeholder groups

✓ Effectively coordinated information platform interoperability between the University of Guelph and Humber College ITAL and designed information dissemination services to meet the unique needs of the Guelph-Humber academic community

✓ Optimized budget expenditures by distributing resources more efficiently within the UofGH budget matrix

✓ Designed and executed a quality service assessment toolkit to continuously evaluate client-oriented information dissemination services

Multilingual Library Services Coordinator, Provincial Library, Ministry of Education, Saskatchewan, Canada, 2006-2008

✓ Provided leadership and strategic planning to re-vamp multilingual information products and services for 11 library systems (332 libraries) across the province

✓ Effectively served as primary contact for activities related to multilingual cultural services for a wide range of stakeholders, including regional library directors, business leaders, government managers, immigration agencies, and culture heritage organizations

✓ Created and led the Multilingual Library Service Committee to build consensus on multilingual service policies between library system managers and representatives from multicultural communities across the province

✓ Researched and reported on the application of social software and electronic information resources to multilingual services

Library Director, Main Library, Universidad Torcuato Di Tella, Buenos Aires, Argentina, 2003-2006

✓ Successfully modernized information services and collections

✓ Performed overall administration and strategic direction of information delivery platforms, including budgeting, space planning, staffing, access policy development, service assessment and fundraising

✓ Spearheaded electronic information resources and digital technologies strategies, including digitization projects, intellectual property management, and digital licensing

✓ Negotiated database subscriptions with vendors on behalf of a cross-national Academic and Research Libraries Consortium involving 8 large institutions

✓ Negotiated affiliation with five philanthropic institutions to sponsor library projects: three from the US (Program for Latin American Libraries and Archives, Latin American Microform Project, and Latin American Open Archives Portal) and two from Argentina (*Fundación Antorchas* and *Centro de Estudios Historicos e Investigacion Parque de España*)

✓ Led data analysis, digitization and cataloguing of newspaper clippings regarding the 2005 Parliamentary Elections in Argentina; provided training workshops to data entry operators

✓ Developed and led the information services assessment program; conducted focus interviews and surveys to assess customer needs

✓ Led the design, development and implementation of the Information Literacy program, including library tours, courses and workshops on information searching; provided specialized reference to faculty in research projects on social sciences, arts and humanities

✓ Led the implementation of *Alephino*, an ExLibris' ALEPH integrated library system software, including the design of the bilingual OPAC-Web interface

✓ Provided customer service training to staff focusing on interpersonal skills development and team-based work

Thought-leadership experience working for the government (in Argentina)

Project Arbiter for Sponsorship Approval, FOMEC (*Fondo para el Mejoramiento de la Calidad de la Enseñanza Universitaria*). Ministry of Education, Argentina, September-October 1999

Responsibility: Assessing a series of library strategic plans submitted by twelve large academic institutions in the nation to the Program FOMEC for sponsorship approval

Academic Library Quality Assessment, CONEAU (*Comisión Nacional para la Evaluación y Acreditación Universitaria*), Ministry of Education, Argentina, April-June 1999

Responsibility: Performing quality assessment of twenty-two academic libraries nationwide for program accreditation with the Ministry