Crowdsource Annotation and Automatic Reconstruction of Online Discussion Threads

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften

vorgelegt von

Emily K. Jamison, M.A.
geboren in Minnesota, USA

Tag der Einreichung: 14. December 2015
Tag der Disputation: 17. February 2016

Referenten: Prof. Dr. phil. Iryna Gurevych, Darmstadt
Prof. Johannes Fürnkranz, PhD, Darmstadt
Prof. Walter Daelemans, PhD, Antwerp

Darmstadt 2016
D17

Abstract

Modern communication relies on electronic messages organized in the form of discussion threads. Emails, IMs, SMS, website comments, and forums are all composed of threads, which consist of individual user messages connected by metadata and discourse coherence to messages from other users. Threads are used to display user messages effectively in a GUI such as an email client, providing a background context for understanding a single message. Many messages are meaningless without the context provided by their thread. However, a number of factors may result in missing thread structure, ranging from user mistake (replying to the wrong message), to missing metadata (some email clients do not produce/save headers that fully encapsulate thread structure; and, conversion of archived threads from over repository to another may also result in lost metadata), to covert use (users may avoid metadata to render discussions difficult for third parties to understand). In the field of security, law enforcement agencies may obtain vast collections of discussion turns that require automatic thread reconstruction to understand. For example, the Enron Email Corpus, obtained by the Federal Energy Regulatory Commission during its investigation of the Enron Corporation, has no inherent thread structure.

In this thesis, we will use natural language processing approaches to reconstruct threads from message content. Reconstruction based on message content sidesteps the problem of missing metadata, permitting post hoc reorganization and discussion understanding. We will investigate corpora of email threads and Wikipedia discussions. However, there is a scarcity of annotated corpora for this task. For example, the Enron Emails Corpus contains no inherent thread structure. Therefore, we also investigate issues faced when creating crowdsourced datasets and learning statistical models of them. Several of our findings are applicable for other natural language machine classification tasks, beyond thread reconstruction.

We will divide our investigation of discussion thread reconstruction into two parts.

First, we explore techniques needed to create a corpus for our thread reconstruction research. Like other NLP pairwise classification tasks such as Wikipedia discussion turn/edit alignment and sentence pair text similarity rating, email thread disentanglement is a heavily class-imbalanced problem, and although the advent of crowdsourcing has reduced anno-

tation costs, the common practice of crowdsourcing redundancy is too expensive for class-imbalanced tasks. As the first contribution of this thesis, we evaluate alternative strategies for reducing crowdsourcing annotation redundancy for class-imbalanced NLP tasks. We also examine techniques to learn the best machine classifier from our crowdsourced labels. In order to reduce noise in training data, most natural language crowdsourcing annotation tasks gather redundant labels and aggregate them into an integrated label, which is provided to the classifier. However, aggregation discards potentially useful information from linguistically ambiguous instances. For the second contribution of this thesis, we show that, for four of five natural language tasks, filtering of the training dataset based on crowdsource annotation item agreement improves task performance, while soft labeling based on crowdsource annotations does not improve task performance.

Second, we investigate thread reconstruction as divided into the tasks of thread disentanglement and adjacency recognition. We present the Enron Threads Corpus, a newly-extracted corpus of 70,178 multi-email threads with emails from the Enron Email Corpus. In the original Enron Emails Corpus, emails are not sorted by thread. To disentangle these threads, and as the third contribution of this thesis, we perform pairwise classification, using text similarity measures on non-quoted texts in emails. We show that i) content text similarity metrics outperform style and structure text similarity metrics in both a class-balanced and class-imbalanced setting, and ii) although feature performance is dependent on the semantic similarity of the corpus, content features are still effective even when controlling for semantic similarity. To reconstruct threads, it is also necessary to identify adjacency relations among pairs. For the forum of Wikipedia discussions, metadata is not available, and dialogue act typologies, helpful for other domains, are inapplicable. As our fourth contribution, via our experiments, we show that adjacency pair recognition can be performed using lexical pair features, without a dialogue act typology or metadata, and that this is robust to controlling for topic bias of the discussions. Yet, lexical pair features do not effectively model the lexical semantic relations between adjacency pairs. To model lexical semantic relations, and as our fifth contribution, we perform adjacency recognition using extracted keyphrases enhanced with semantically related terms. While this technique outperforms a most frequent class baseline, it fails to outperform lexical pair features or tf-idf weighted cosine similarity. Our investigation shows that this is the result of poor word sense disambiguation and poor keyphrase extraction causing spurious false positive semantic connections.

Publications of the contributions are listed in Section 1.3. Figure 1.1 shows an overview of the topics of the contributions and how they are interrelated.

In concluding this thesis, we also reflect on open issues and unanswered questions remaining after our research contributions, discuss applications for thread reconstruction, and suggest some directions for future work.

Zusammenfassung

Moderne Kommunikation beruht auf elektronischen Nachrichten, die in Form von Threads organisiert sind. E-Mails, Sofortnachrichten, SMS, Kommentare auf Webseiten und in Foren sind aus solchen Threads aufgebaut - diese wiederum bestehen aus einzelnen Benutzernachrichten, die mithilfe von Metadaten verbunden sind und zwischen denen Diskurskohärenz besteht. Threads werden benutzt, um Benutzernachrichten effektiv in einer GUI, wie etwa einem E-Mail-Programm, zu visualisieren. Sie stellen also einen Hintergrundkontext bereit, ohne den einzelne Nachrichten oft nicht verstanden werden können. Allerdings kann es durch eine Reihe von Faktoren dazu kommen, dass eine solche Thread-Struktur verloren geht: Angefangen von Benutzerfehlern (z.B. dem Antworten auf eine falsche Nachricht), über fehlende Metadaten (manche E-Mail-Programme erzeugen E-Mail-Header, die nicht die volle Thread-Struktur enthalten; auch Konvertierungen von alten Threads können in fehlenden Metadaten resultieren) bis hin zu absichtlich verschleierter Struktur (etwa durch Benutzer, die es Dritten erschweren wollen, eine Diskussion nachzuvollziehen, und dazu Metadaten vermeiden oder entfernen). Im Bereich Sicherheit benötigen Strafverfolgungsbehörden daher eine automatische Thread-Rekonstruktion, um große Mengen an gesammelten elektronischen Nachrichten aus Diskussionen verstehen zu können. Beispielsweise besitzt das Enron Email Corpus, das von der Federal Energy Regulatory Commission der USA während der Ermittlungen beim Energiekonzern Enron zusammengetragen wurde, keine inhärente Thread-Struktur.

In dieser Arbeit verwenden wir Ansätze aus der maschinellen Sprachverarbeitung (Natural Language Processing, NLP), um Threads aus Nachrichteninhalten zu rekonstruieren. Eine solche Rekonstruktion basierend auf den Inhalten umgeht das Problem fehlender Metadaten und erlaubt eine nachträgliche Restrukturierung und damit auch ein Verstehen der gesamten Diskussion. Wir untersuchen Korpora bestehend aus E-Mail-Threads und Wikipedia-Diskussionen. Allerdings herrscht eine Knappheit an geeigneten, annotierten Korpora. Zum Beispiel enthält das Enron Emails Corpus keine Angaben zur Thread-Struktur. Aus diesem Grund erforschen wir außerdem Probleme, die beim Erstellen von crowdgesourcten Datensätzen und beim Trainieren maschineller Lernverfahren auf solchen Datensätzen auftreten.

Viele unserer Ergebnisse sind daher über die Thread-Rekonstruktion hinaus auch auf andere automatische Klassifizierungsaufgaben für natürliche Sprache anwendbar.

Wir gliedern unsere Erforschung der Rekonstruktion von Diskussions-Threads in zwei Teile auf.

Zuerst untersuchen wir Methoden für die Erstellung eines Korpus, das der Forschung an Thread-Rekonstruktion dienen soll. Wie andere Problemstellungen im Bereich paarweiser Klassifikation in NLP, etwa die Textähnlichkeitsbewertung für Satzpaare oder das Alignment von Sprecherwechseln in Wikipedia-Diskussionen zu Artikeländerungen, ist auch die Wiederherstellung von E-Mail-Threads ein stark klassen-unbalanciertes Problem. Trotz des Aufkommens von Crowdsourcing, das Annotationskosten deutlich reduziert hat, ist die bisher übliche Praxis der Crowdsourcing-Redundanz zu teuer für Aufgaben mit Klassen-Ungleichgewicht. Als ersten Beitrag dieser Arbeit evaluieren wir alternative Strategien, um Crowdsourcing-Redundanz für Annotationen in klassen-unbalancierten NLP Aufgaben zu reduzieren. Wir untersuchen außerdem Methoden, den bestmöglichen maschinellen Klassifikator auf unseren crowdgesourcten Labeln zu trainieren. Um Rauschen in Trainingsdaten zu reduzieren, sammeln die meisten Crowdsourcing-Annotationsexperimente in NLP mehrere redundante Label und aggregieren sie zu einem ganzheitlichen Label, das dann an den Klassifikator weitergegeben wird. Allerdings verwirft solch eine Aggregation potenziell nützliche Informationen von linguistisch ambigen Instanzen. Für den zweiten Beitrag dieser Arbeit zeigen wir für vier von fünf NLP-Problemstellungen, dass das Filtern von Trainingsdaten basierend auf Inter-Annotator-Agreement von Instanzen die Effektivität des Klassifikators steigern kann, im Gegensatz zu Soft-Labeling, das keine Ergebnisverbesserungen liefert.

Zweitens untersuchen wir Thread-Rekonstruktion, aufgeteilt in die Entflechtung von Threads und die Erkennung von Adjazenz. Wir stellen das Enron Threads Corpus vor, ein neu extrahiertes Korpus von 70.178 Threads, bestehend aus jeweils mehreren E-Mails des Enron E-mail Corpus. Die E-Mails im ursprünglichen Enron Emails Corpus sind nicht nach Threads sortiert. Um Threads zu finden und zu entflechten, wenden wir als dritten Beitrag dieser Arbeit paarweise Klassifikation an. Dazu benutzen wir Textähnlichkeitsmaße auf nicht-zitiertem Text in E-Mails. Wir zeigen zweierlei: i) Textähnlichkeitsmaße, die auf dem Textinhalt operieren, übertreffen stil- und strukturorientierte Maße sowohl in klassen-balancierten als auch in klassen-unbalancierten Experimenten. Und ii) obwohl die Effektivität der Features von der semantischen Ähnlichkeit des Korpus abhängt, sind inhaltliche Features auch dann effektiv, wenn die semantische Ähnlichkeit kontrolliert wird. Um Threads zu rekonstruieren ist es zusätzlich notwendig, Adjazenzbeziehungen zwischen Paaren zu identifizieren. Für die Wikipedia-Diskussionen sind keine Metadaten verfügbar; außerdem sind Dialogakt-Typologien, die für andere Domänen hilfreich sein können, hier nicht nutzbar. Als vierten Beitrag zeigen wir anhand unserer Experimente, dass die Erkennung von Adjazenzpaaren unter Benutzung von „Lexical-Pair-Features“ durchgeführt werden kann. Dieser Ansatz ist robust auch bei Berücksichtigung von Topic Bias der Diskussionen und benötigt weder Dialogakt-Typologie noch

Metadaten. Allerdings bilden Lexical-Pair-Features nicht tatsächlich die lexikalisch-semantischen Relationen zwischen Adjazenzpaaren ab. Um also lexikalisch-semantische Beziehungen zu modellieren, führen wir als unseren fünften Beitrag Adjazenz-Erkennung mittels extrahierter Keyphrases durch, die mit semantisch ähnlichen Termen angereichert werden. Diese Methode liefert bessere Ergebnisse als eine „Most-Frequent-Class-Baseline“, zeigt aber keine Verbesserung gegenüber Lexical-Pair-Features oder mittels Tf-idf gewichteter Kosinus-Ähnlichkeit. Unsere Untersuchung zeigt, dass dies das Resultat fehlerhafter Word-Sense-Disambiguation und Keyphrase-Extraction ist, was falsche semantische Verbindungen hervorbringt.

Publikationen, die unsere Beiträge behandeln, sind in Section 1.3 aufgelistet. Figure 1.1 zeigt einen Überblick über die Themen der Beiträge und wie sie miteinander in Beziehung stehen.

Abschließend besprechen wir nach den vorliegenden Beiträgen verbleibende ungelöste Probleme und offene Fragen, diskutieren Anwendungen für Thread-Rekonstruktion und zeigen mögliche Wege für weiterführende Arbeiten auf.

(Dieses Abstract wurde aus dem Englischen übersetzt von Erik-Lân Do Dinh.)

Acknowledgements

A dissertation is a group effort, and many people have assisted with this one. I would like to thank Prof. Dr. Iryna Gurevych for affording me the opportunity to conduct this research, for constructing a supportive community of researchers to enable this research, and for providing efficient supervision. I am grateful for Prof. Dr. Johannes Fürnkranz and Prof. Dr. Walter Daelemans for finding the time to evaluate this thesis. This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Center for Advanced Security Research (`www.cased.de`). Additionally, some prior work has been conducted at Alias-i, with thanks to Dr. Breck Baldwin.

I would like to thank the researchers who have taken the time to either assist me with specific research problems or to read my papers and provide feedback on work discussed in this thesis, including but not limited to, Prof. Dr. Chris Biemann, Prof. Dr. Ulf Brefeld, Dr. Bob Carpenter, Dr. Kostadin Cholakov, Dr. Johannes Daxenberger, Dr. Richard Eckart de Castilho, Dr. Oliver Ferschke, Lucie Flekova, Dr. Ivan Habernal, Silvana Hartmann, Dr. Michael Matuschek, Dr. Christian M. Meyer, Pedro Santos, Christian Stab, Dr. György Szarvas, Prof. Dr. Torsten Zesch, and Hans-Peter Zorn. I greatly appreciate the brainstorming sessions with our many UKP guest visitors, and paper polishing from the anonymous reviewers. I would like to thank the Special Interest Groups DKPro Text Classification and DKPro Core for their community support in developing software used in this thesis, as well as the IT Forensics group of CASED for discussion on the intersection of NLP and IT Security. Additionally, I am grateful to Ilya Kuznetsov for annotation, Erik-Lân Do Dinh for translation, and the friendly UKP System-Admin team for many hours of support.

Approximately 76 researchers have worked at UKP and LangTech during my tenure, and I am so lucky and thankful to have had fascinating lunchtime research discussions with nearly each and every one of you.

I would specifically like to thank Dr. Yannick Versley for many, many years of NLP research encouragement and feedback. Finally, I would like to thank Dr. Graham King for enthusiasm and encouragement of my research journey.

Contents

Foreword

On June 1, 1880, Mark Twain (1880) published a short story in the magazine, *The Atlantic*. The story featured an odd linguistic phenomenon created by emerging technology (i.e., the telephone): short pieces of discussion that were not understandable without the rest of the discussion. An excerpt from the story is reprinted below.

> A TELEPHONIC CONVERSATION (excerpt)
>
> Without answering, I handed the telephone to the applicant, and sat down. Then followed that queerest of all the queer things in this world—a conversation with only one end of it. You hear questions asked; you don't hear the answer. You hear invitations given; you hear no thanks in return. You have listening pauses of dead silence, followed by apparently irrelevant and unjustifiable exclamations of glad surprise or sorrow or dismay. You can't make head or tail of the talk, because you never hear anything that the person at the other end of the wire says. [...]
>
> Yes? Why, how did THAT happen?
>
> Pause.
>
> What did you say?
>
> Pause.
>
> Oh no, I don't think it was.
>
> Pause.
>
> NO! Oh no, I didn't mean THAT. I meant, put it in while it is still boiling—or just before it COMES to a boil.
>
> Pause.
>
> WHAT?
>
> Pause.
>
> I turned it over with a backstitch on the selvage edge.
>
> Pause.
>
> Yes, I like that way, too; but I think it's better to baste it on with Valenciennes or bombazine, or something of that sort. It gives it such an air—and attracts so much noise.
>
> Pause.

It’s forty-ninth Deuteronomy, sixty-forth to ninety-seventh inclusive. I think we ought all to read it often.

Pause.

Perhaps so; I generally use a hair pin.

Pause.

What did you say? (ASIDE.) Children, do be quiet!

Pause

OH! B FLAT! Dear me, I thought you said it was the cat!

Pause.

Since WHEN?

Pause.

Why, I never heard of it.

Pause.

You astound me! It seems utterly impossible!

Pause.

WHO did?

Pause.

Good-ness gracious!

Pause.

Well, what IS this world coming to? Was it right in CHURCH?

Pause.

And was her MOTHER there?

Pause.

Why, Mrs. Bagley, I should have died of humiliation! What did they DO?

Long pause.

I can’t be perfectly sure, because I haven’t the notes by me; but I think it goes something like this: te-rolly-loll-loll, loll lolly-loll- loll, O tolly-loll-loll-LEE-LY-LI-I-do! And then REPEAT, you know.

Pause.

Yes, I think it IS very sweet—and very solemn and impressive, if you get the an-dantino and the pianissimo right.

Pause.

Oh, gum-drops, gum-drops! But I never allow them to eat striped candy. And of course they CAN’T, till they get their teeth, anyway.

Pause.

WHAT?

Pause.

Oh, not in the least—go right on. He’s here writing—it doesn’t bother HIM.

Pause.

Very well, I’ll come if I can. (ASIDE.) Dear me, how it does tire a person’s arm to hold this thing up so long! I wish she’d—

```
Pause.

Oh no, not at all; I LIKE to talk—but I'm afraid I'm keeping you from your affairs.

Pause.

Visitors?

Pause.

No, we never use butter on them.

Pause.

Yes, that is a very good way; but all the cook-books say they are very unhealthy
when they are out of season. And HE doesn't like them, anyway—especially canned.

Pause.

Oh, I think that is too high for them; we have never paid over fifty cents a bunch.

Pause.

MUST you go? Well, GOOD-by.

Pause.

Yes, I think so. GOOD-by.

Pause.

Four o'clock, then—I'll be ready. GOOD-by.

Pause.

Thank you ever so much. GOOD-by.

Pause.

Oh, not at all!—just as fresh—WHICH? Oh, I'm glad to hear you say that. GOOD-by.

(Hangs up the telephone and says, "Oh, it DOES tire a person's arm so!")
```

The dawning of the telephone age permitted real-time dialogue whose comprehensibility relied on access to the entire discussion, and yet whose turns could be overheard without such a context. Twain listened to the turns of one speaker, yet could not hear the responses of the other. The discussion made sense to the participants, but was meaningless to the third-party observer.

How can one make sense of a discussion with missing turns? What linguistic information can be utilized from the existing turns to predict the contents of the missing ones?

Twain was early to ponder this problem, and likely had no idea how common it would become: in modern day, nearly all of our web-based discussion, from emails to IRC chats, to forums and discussion boards, to article responses, relies on the structure of the discussion to render the dialogue comprehensible and informative. Without knowing which turn is a reply to which other turn, dialogue becomes as mysterious as this story.

In this dissertation, we confront the problem of incomprehensible isolated discussion turns. Specifically, we work towards putting together the pieces of broken discussion: thread reconstruction.

Chapter 1

Introduction

Modern communication relies on electronic discussion. In 2010, an estimated 32B non-spam emails were sent per day (Radicati and Levenstein, 2013). In 2013, 400M comments and 3.6B comment votes were posted to 40M discussions on the social voting site Reddit (Grant, 2013). In 2012, 17.6B SMS messages were sent per day (strategyeye.com, 2014). Widespread adaptation of electronic communication in the 1990's has resulted in vast numbers of online discussions, known as threads. Emails, IMs, SMS, website comments, and forums are all composed of *discussion threads*, which consist of individual *turns* (user messages) connected by *metadata* (i.e. subject line, reply-to id) and discourse coherence to messages from other users. Threads are used to display user messages effectively in a GUI such as an email client or forum website, providing necessary context for understanding a single message.

The ubiquity of electronic messaging and discussion threads has rendered thread organization more important than ever before. A discussion cannot be fully understood if its individual messages are missing, out of order, or are mixed with the messages of other threads. However, a number of factors may result in missing thread structure, from user mistake (replying to the wrong message), to missing metadata (some email clients do not produce/save headers that fully encapsulate thread structure; and, conversion of archived threads from over repository to another may also result in lost metadata), to covert use (users may not use metadata to make discussions difficult for 3rd parties to understand). When metadata is missing, the only solution is to reconstruct the thread structure based on the linguistic content of the messages.

In this dissertation, we investigate this problem of thread reconstruction as an NLP task of reconstructing threads using message content alone. Additionally, as an investigation of an under-researched NLP task, we explore crowdsourcing processes necessary to create and use the novel datasets required for discussion thread reconstruction. Thus, this thesis is divided into two parts. In Part I, we discuss our crowdsourcing contributions for corpus annotation and machine learning on crowdsourced labels, towards a goal of corpus production for thread reconstruction experiments. In Part II, we provide background on discussion thread recon-

struction, and we report results of our thread reconstruction experiments, as divided into the subtasks *thread disentanglement* (separating intermixed turns by their discussions) and *adjacency recognition* (identifying reply-to relations among pairs of turns). An overview of these topics can be seen in Figure 1.1.

1.1 Motivations

In this section we discuss high-level research questions addressed by this thesis. Additionally, individual chapters discuss specific research questions relevant to those chapters. The greater applicability of our findings, beyond the immediate question being addressed, is also discussed in the individual chapters.

We are interested in the following high-level thread reconstruction research questions, which we expect will have real-world use:

- Given an unordered bag of discussion turns,
  - how can we divide the turns by their discussions?
  - how can we assign discussion relations (i.e., reply-to relations) between the turns?
- Given an unverified discussion thread,
  - how can we identify incorrect discussion relation links between the turns, and suggest better ones?

Additionally, an under-investigated task such as thread reconstruction needs new corpora:

- What processes should be used to obtain and learn a model from a crowdsource-labeled corpus?

These are fundamental questions whose complete answers lie outside the scope of this thesis. However, these questions motivate the specific research questions addressed in Section 1.2. Eventually, progress on these high-level questions will contribute to the following applications, discussed in detail in Chapter 9.2.

- Greater access to human-annotated corpora for the NLP research community.
- Better machine learning models trained on crowdsourced datasets.
- The ability to reconstruct discussion threads via only message content, with benefits such as:
  - Improved evidence collection by law enforcement
  - Thread manipulation detection for law enforcement
  - Email clients that provide better thread organization and display
  - Email clients and forum software that provide real-time user suggestion/correction

  - Improved email client ad search
  - Improved chatbot discourse generation
  - Software for post-hoc forum, website comment, and Wikipedia discussion thread structure correction

In fact, all of the many varied forms of modern electronic communication, from emails to internet relay chats to Wikipedia discussion pages to social voting sites to news article comment sections to question-answering websites, would all be impacted by technology to identify and track the structure of the discussion. With the increasing ubiquity of electronic text communication, this technological need is becoming more relevant than ever.

1.2 Contributions and Findings

Previously in Section 1.1, we have presented several of the high-level research questions motivating this thesis. In this section, we summarize our most important findings and contributions. We also summarize our novel tasks, corpora, methods, and concepts.

Our findings concerning crowdsourcing processes necessary to create and use the novel datasets required for discussion thread reconstruction:

- Thread reconstruction requires the determination of relations between pairs of discussion turns. A corpus made of these pairs of turns is *class-imbalanced*, because the negative and positive classes of pairs have very different prior probability distributions. We show that annotation of a class-imbalanced dataset can be very expensive, due to extraneous annotations of unneeded common-class (high prior probability class) instances while searching for rare-class (low prior probability class) instances, and we investigate techniques to reduce these annotation costs. In an investigation of annotation affordability, and using three class-imbalanced corpora, we showed that annotation redundancy for noise reduction is expensive on a class-imbalanced dataset, and should be discarded for instances receiving a single common-class label. We also showed that this simple technique, which does not require any training data, produces annotations at approximately the same cost of a metadata-trained, supervised cascading machine classifier, or about 70% cheaper than five-vote majority-vote aggregation. We expect that future work will combine this technique for seed data creation with algorithms such as Active Learning to create corpora large enough for machine learning, at a reduced cost.

- The crowdsource labeling of a thread reconstruction dataset will typically result in redundant labels for each instance, so we have explored different techniques to learn the best classifier model from the redundant labels. In an investigation of machine learning with crowdsource-annotated datasets, for five natural language tasks, we examined the impact of informing the classifier of *item agreement*, by means of *soft label-*

ing (the assignment of multiple partial labels to the same machine learning *instance*) and low-agreement training instance *filtering* (removing certain instances from a training dataset). We found a statistically significant benefit from filtering in four of our five tasks, and strongest improvements for linguistically ambiguous cases. The non-benefited task, stemming, had the lowest number of *item agreement categories* of the five tasks, preventing fine-grained agreement training filtering, which explains why filtering showed no benefit. However, we also observed our training strategies impacted some classification categories more than others, increasing sample-selection bias, which negatively impacts model learning. Our findings suggest that the best crowdsource label training strategy is to remove low item agreement instances, although care must be taken to reduce sample-selection bias.

Our findings regarding thread reconstruction:

- We investigated the use of different types of text similarity features for the pairwise classification of emails for *thread disentanglement*. We found that content similarity features are more effective than style or structural features across class-balanced and class-imbalanced environments. There appear to be more stylistic features of the text uncaptured by our similarity metrics, which humans access for performing the same task. We have shown that semantic differences between corpora will impact the general effectiveness of text similarity features, but that content features remain effective.

- In an evaluation of the use of lexical pairs for *adjacency recognition*, we have shown that lexical pairs are helpful, outperforming cosine similarity. We have further shown that this benefit is robust to topic bias control. Our error analysis raises intriguing questions for future research, showing that a number of forms of deeper linguistic analysis, such as centering theoretic analysis, stance detection, and lexical semantic modeling may be necessary to reduce the current error rate in metadata-less adjacency recognition.

- We evaluated the use of lexical expansion as a source of knowledge-rich features for adjacency recognition. We found that, despite the intuitive appeal of lexical expansion to represent the topic of a text, lexical expansion fails to outperform simple knowledge-poor approaches such as tf-idf cosine similarity and lexical pairs. Additionally, in a comparison of nouns versus keyphrases as terms to be expanded, we found that choice depends on which machine learning features are used, as well as corpus frequency of names and jargon.

The contributions summarized above can be alternatively characterized as novel concepts, tasks, methods, and corpora.

The *concepts* introduced and investigated in this work include:

- *Crowdsource annotation of a class-imbalanced dataset* as a financial/resource concern
- *Machine learning on crowdsourced labels* affected by informational properties of crowdsourced labels that remain constant across different natural language tasks
- *Lexical pairs* as an effective feature-space-reduction from SVM feature auto-combination

In our work, we propose several new *tasks*.

- *Metadata-free thread reconstruction*, thread disentanglement, and adjacency recognition
- Email thread reconstruction from quoted email material via *clustering and segmentation of quoted emails*
- Pairwise thread disentanglement as a *content, structural, and style-based text similarity problem*

To solve these tasks, we conduct experiments using the following new *methods*.

- Automatic adjacency and disentanglement pairwise classification of *emails* using only *message-content features*
- Automatic adjacency and disentanglement pairwise classification of *Wikipedia discussion turns* using only *message-content features*
- Automatic adjacency recognition using *knowledge-rich features*
- *Classifier cascade of crowdsourced datasets* using only metadata features
- *Rule-based cascade of crowdsourced datasets* using only metadata features

Experiment design is critical to produce meaningful results. In our experiments, we propose the following techniques.

- Controlling of semantic similarity during evaluation of text similarity for email thread disentanglement
- Prevention of information leak between discussion turn pairs from the same discussion thread
- Most-frequent-class baseline calculation that is sensitive to entropy of class balance of source discussion threads

For the work described in this thesis, we produced two corpora, as summarized in Table 1.1. Other corpora used in our experiments are summarized in Table 1.2. The tables provide a reference for the acronyms used throughout this work.

Task-specific research questions motivating the work in Chapter 3, Chapter 4, Chapter 6, Chapter 7, and Chapter 8 are listed at the beginning of their respective chapters.

[1] `https://www.ukp.tu-darmstadt.de/data/text-similarity/email-disentanglement`

[2] Some corpus info is here: `https://www.ukp.tu-darmstadt.de/data/text-similarity/re-rating-studies`

Corpus	Full name	Purpose	Self-created	Availability and License	Described in Chapter
ETC	Enron Threads Corpus	Thread disentanglement and adjacency recognition	yes	yes, public domain[1]	5
ECD	Enron Crowdsourced Dataset	Crowdsourcing experiments	yes	yes, public domain[1]	2

Table 1.1: Self-produced corpora created and used as contributions in this thesis. All corpora are in English.

1.3 Publication Record

We have published the major contributions of this thesis in peer-reviewed conference or workshop proceedings of major events in natural language processing. The chapters extending these publications are indicated below. A full list of our publications is available in the appendix.

Emily K. Jamison and Iryna Gurevych: 'Noise or additional information? Using crowdsource annotation item agreement for natural language tasks', in: *Proceedings of EMNLP 2015: Conference on Empirical Methods in Natural Language Processing* (EMNLP 2015), p. 291–297, Lisbon, Spain, 2015. (Chapter 4)

Emily K. Jamison and Iryna Gurevych: 'Adjacency Pair Recognition in Wikipedia Discussions using Lexical Pairs', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing* (PACLIC), p. 479–488, Phuket, Thailand, 2014. (Chapter 7)

[3] http://www.ukp.tu-darmstadt.de/data/wikidiscourse

[4] https://sites.google.com/site/amtworkshop2010/data-1

[5] http://www.pascal-network.org/Challenges/RTE/Datasets/

[6] http://sites.google.com/site/nlpannotations/

[7] http://www.ark.cs.cmu.edu/TweetNLP

[8] http://www.lowlands.ku.dk/results/

[9] http://nlp.cs.swarthmore.edu/semeval/tasks/

[10] http://sites.google.com/site/nlpannotations/

[11] Self-created with Bob Carpenter and Breck Baldwin

[12] http://github.com/bob-carpenter/anno

[13] I created the Human Intelligence Task (HIT) Layout and Setup for the crowdsource annotation of this corpus. As per agreement with Johannes Daxenberger, ETP-GOLD and corpus annotation guidelines are contributions of only JD's dissertation.

[14] http://www.upk.tu-darmstadt.de/data/edit-turn-pairs

Corpus	Full name	Purpose	Self-created	Availability and License	Described in Chapter
SENT-PAIRS	30 Sentence Pairs	Crowdsourcing experiments	no	yes, upon request[2]	2
EWDC	English Wikipedia Discussions Corpus	Adjacency pair recognition	no	yes, CC-by-SA[3]	2
YANO2010	American Political Blog Post Corpus	Biased language detection	no	yes, license unknown[4]	4
PASCAL RTE-1	PASCAL Recognizing Textual Entailment Dataset-1	Recognizing textual entailment	no	yes, no license[5]	4
RTEANNO	Snow et al. 2008's MTurk Annotations for PASCAL RTE-1	Recognizing textual entailment	no	yes, no license[6]	4
GIMBEL2011	Gimbel et al. 2011's POS Twitter Corpus	POS tagging	no	yes, by CC-BY[7]	4
GIMBELANNO	Hovy et al. 2014's Crowdflower Annotations for GIMBEL2011	POS tagging	no	yes, no license[8]	4
SEM2007	SemEval 2007 Affective Text Task	Affect recognition	no	yes, no license[9]	4
SEMANNO	Snow et al. 2008's MTurk Annotations for SEM2007	Affect recognition	no	yes, no license[10]	4
CARP2009	MTurk Stems Corpus	Morphological stemming	joint[11]	yes, by BSD-Simplified[12]	4
ETP-GOLD	Wikipedia Edit-Turn-Pair Corpus	Crowdsourcing experiments	no[13]	yes, CC-by-SA[14]	2

Table 1.2: Other corpora used in this thesis. All corpora are in English.

Emily K. Jamison and Iryna Gurevych: 'Needle in a Haystack: Reducing the Costs of Annotating Rare-Class Instances in Imbalanced Datasets', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing* (PACLIC), p. 244-253, Phuket, Thailand, 2014. (Chapter 3)

Emily K. Jamison and Iryna Gurevych: 'Headerless, Quoteless, but not Hopeless? Using Pairwise Email Classification to Disentangle Email Threads', in: *Proceedings of 9th Conference on Recent Advances in Natural Language Processing* (RANLP 2013), p. 327–335, Hissar, Bulgaria, 2013. (Chapters 5 and 6)

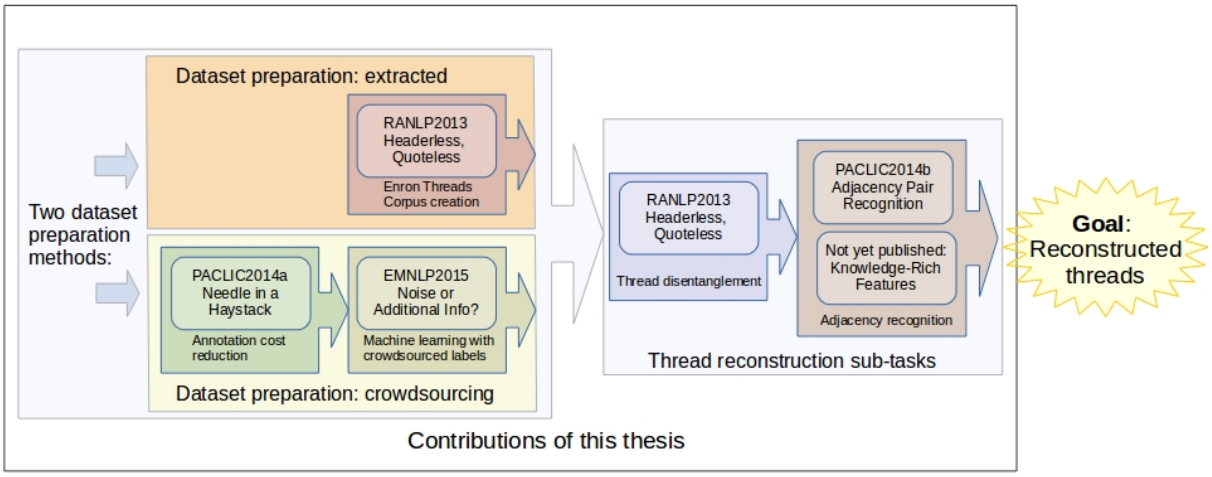

Figure 1.1: An overview of tasks and publications of this thesis.

1.4 Thesis Outline and Term Conventions

This section provides an overview of the organization of this thesis. This thesis presents our contributions to discussion thread reconstruction, as well as our contributions to crowdsourcing processes necessary to create and use the novel datasets required for discussion thread reconstruction. Thus, this thesis is divided into two parts. In the first part, we discuss our crowdsourcing contributions for corpus annotation and machine learning on crowdsourced labels, towards a goal of corpus production for thread reconstruction experiments. In the second part, we provide background on discussion thread reconstruction, and we present our discussion thread reconstruction contributions. Figure 1.1 provides an overview of this thesis.

Part I In the first part, we explore techniques needed to create a corpus for our thread reconstruction research. We investigate crowdsourcing as a cheap and effective source of data annotations.

In Chapter 2, we provide background on the practice of crowdsource annotation. This includes an overview of crowdsourcing including its various forms, a brief history of crowdsourcing, demographic statistics, tasks that have been successfully crowdsourced, economic issues of the crowdsource labor market, and problems with label quality.

In Chapter 3, we present our contributions on class-imbalanced crowdsource corpus creation. Like other NLP pairwise classification tasks such as Wikipedia discussion turn/edit alignment and sentence pair text similarity rating, many thread reconstruction tasks such as email thread disentanglement are heavily class-imbalanced problems, and although the advent of crowdsourcing has reduced annotation costs, common practice of crowdsourcing redundancy is too expensive for class-imbalanced tasks. We evaluate alternative strategies for

reducing crowdsourcing annotation redundancy for class-imbalanced NLP tasks. Our findings are needed for reducing the cost of annotating thread reconstruction corpora, yet are additionally applicable to other class-imbalanced corpus annotation, such as the tasks examined in this chapter.

A crowdsource-annotated thread reconstruction corpus will have redundant labels that must be converted into gold standard annotations. In Chapter 4, we present our contributions on comparing techniques to learn the best machine classifier from crowdsourced labels. In order to reduce noise in training data, most natural language crowdsourcing annotation tasks gather redundant labels and aggregate them into an integrated label, which is provided to the classifier. However, aggregation discards potentially useful information from linguistically ambiguous instances. We show that, for four of five natural language tasks, filtering of the training dataset based on crowdsource annotation item agreement improves task performance, while soft labeling based on crowdsource annotations does not improve task performance. Our findings are needed for learning the best classifier from a crowdsource-annotated thread reconstruction corpus, yet are additionally applicable to the wide range of text classification tasks investigated in the chapter.

Part II In the second part, we investigate thread reconstruction as divided into the tasks of thread disentanglement and adjacency recognition.

In Chapter 5, we provide background on discussion thread reconstruction, and we discuss types of online discussion threads and threading problems. We present the *Enron Threads Corpus* (ETC), a newly-extracted corpus of 70,178 multi-email threads with emails from the Enron Email Corpus. We also give an overview of the other major corpus we used in our thread reconstruction experiments, the *English Wikipedia Discussions Corpus* (EWDC).

In Chapter 6, we present our contributions on email thread disentanglement. In the original Enron Emails Corpus, emails are not sorted by thread. To disentangle these threads, we perform pairwise classification, using text similarity measures on non-quoted texts in emails. We show that i) content text similarity metrics outperform style and structure text similarity metrics in both a class-balanced and class-imbalanced setting, and ii) although feature performance is dependent on the semantic similarity of the corpus, content features are still effective even when controlling for semantic similarity.

In Chapter 7, we present our contributions on adjacency recognition. To reconstruct threads, it is necessary to identify adjacency relations among pairs. For the forum of Wikipedia discussions, metadata is not available, and dialogue act typologies, helpful for other domains, are inapplicable. Via our experiments, we show that adjacency recognition can be performed using lexical pair features, without a dialogue act typology or metadata, and that this is robust to controlling for topic bias of the discussions.

In Chapter 8, we present our contributions on the use of lexical expansion for adjacency pair recognition. Lexical pair features do not effectively model the lexical semantic relations

between adjacency pairs. To model lexical semantic relations, we perform adjacency recognition using extracted keyphrases enhanced with semantically related terms. While this technique outperforms a most frequent class baseline, it fails to outperform lexical pair features or tf-idf weighted Cosine Similarity. Our investigation shows that this is the result of poor word sense disambiguation and poor keyphrase extraction causing spurious false positive semantic connections.

Typography and Terminology We wish to note the following typographical practices. Important terms are printed in *italics* at the place where they are introduced or re-introduced. All corpora used in this thesis are named in Small Capital letters.

The following important terms are explained in their respective sections of the thesis, as well as here for convenience.

Class imbalance refers to severely unequal prior class probabilities in the dataset.

Common-class instances are members of a class with high prior class probability.

Rare-class instances are members of a class with low prior class probability.

Item agreement is the inter-annotator agreement between labels for one machine learning *instance*, such as one token with five different POS-tags assigned by five different expert annotators.

Soft labeling is the assignment of multiple partial labels to the same machine learning instance, such that the weight of all soft labels of an instance adds up to 1.0; in our work, soft labeling is practiced on a training dataset but a classifier outputs hard labels on the evaluation dataset.

Filtering is the practice of removing certain instances from a training dataset under certain conditions of the instance, such as low *inter-annotator agreement* of labels on that instance, for machine learning.

A *discussion* is a conversation between two or more *participants* (people), in which the participants participate for a total of at least two discussion turns.

A *discussion turn* is an uninterrupted utterance by a participant; examples include a single email or a single forum post.

Thread reconstruction is the general task of assigning relations between discussion turns to organize the turns by discussion and by *reply-to relations* (relations where one turn is in response to another turn).

Two subtasks of thread reconstruction include *thread disentanglement*, which identifies which discussion a turn belongs to, and *adjacency recognition*, which identifies reply-to relations among pairs of turns.

More terminology definitions are available on the topic of crowdsourcing in Chapter 2 and on the topic of thread reconstruction in Chapter 5.

Part I

Experiments in Crowdsourcing Annotation

Chapter 2

Crowdsourcing NLP Annotation

In this thesis, we investigate the NLP task of discussion-thread reconstruction. As a subdomain of NLP, this task has received less attention than other subdomains, and it lacks resources. There are few publicly available corpora for studying thread reconstruction. Thus, as frequently happens with understudied NLP tasks, we must build our own corpora.

In Part I of this thesis, we investigate crowdsourcing techniques that are relevant to building and using a crowdsourced corpus for thread reconstruction. These crowdsourcing techniques are also relevant for many NLP tasks beyond thread reconstruction. We discuss broader applicability with each investigation.

In this chapter, we provide a background on crowdsourcing. Section 2.1 contains an overview on crowdsourcing, and Section 2.2 discusses various types of crowdsoucing. A brief history of crowdsourcing is provided in Section 2.3. In Section 2.4, we discuss demographics of crowdsource workers, and in Section 2.5 we discuss what annotations tasks have been successful with crowdsourcing. In Section 2.6, we discuss economic issues of the crowdsource labor market. Finally, in Section 2.7, we discuss problems with crowdsourcing label quality, including spam/worker fraud, worker mistakes (accidental/random), worker quality (systemic), worker bias, and the data-driven problem of ambiguity.

2.1 What is Crowdsourcing?

Crowdsourcing is the use of anonymous persons from the internet to solve specific, internet-based, human-skill tasks. It is an alternative to hiring and training in-house employees, a long-term solution that may not be suitable for short or small tasks.

Yuen et al. (2011a) describes crowdsourcing as a "distributed problem-solving and business production model," one that "makes use of human abilities for computation to solve problems" (Yuen et al., 2009). Crowdsourcing commonly refers to a micro-task market (Kittur et al., 2008), an online system of very small payments for very brief work tasks, using frameworks

such as Amazon's *Mechanical Turk*[15], *CrowdFlower*[16], or *Taskcn*[17]. The tasks are of a nature that requires human intelligence, such as identifying objects in photos or translating product descriptions. The tasks require little time or expertise, so they offer little compensation per task, frequently USD$0.01 to USD$0.10. The benefit of crowdsource platforms is "the capacity to organize people into economically productive units" (Howe, 2008).

Jeff Howe (2008) described crowdsourcing from a labor market perspective: " 'Crowdsourcing' is the act of taking a task traditionally performed by a designated agent (such as an employee or contractor) and outsourcing it by making an open call to an undefined but large group of people." From this perspective, a wider definition of crowdsourcing refers to many collaboratively-built Web 2.0 platforms: *Wikipedia*[18], *Yahoo! Answers*[19], Yahoo's *flickr*[20], the social bookmarking sites *del.icio.us*[21] and *Reddit*[22], to social games such as *TagATune*[23] and *Google Image Labeler*[24] to creative systems such as the *Sheep Market*[25] (Yuen et al., 2011a). *YouTube*[26] demonstrates a reward-based participation market: when a contributor's videos receive less attention, the contributor posts fewer new videos, which results in less attention, until frequently the user stops contributing (Huberman et al., 2009). In addition, variants of the micropayments-for-work-tasks model exist on a larger scale: InnoCentive[27] and NineSigma[28] allow requesting companies to post research and development work tasks, and workers are rewarded $10,000 or $25,000 for inventing a solution for the task, whose in-house development would have cost several times as much, if it was solved at all (Howe, 2006).

A variant of crowdsourcing in which multiple workers submit solutions for single task is the *Witkey* labor market (Yang et al., 2008). Witkeys are popular in China and include websites such as Taskcn.com, zhubajie.com, and k68.cn. They are named after the first website to use the model in 2005, Witkey.com. In the Witkey model, the website hosts tasks from requesters, and multiple workers submit solutions for each task. The requester chooses the best solution and pays the reward to that submitter. The website, like most crowdsourcing websites, keeps a small portion of the reward in exchange for hosting the task. DiPalantino and Vojnovic (2009) investigated the reward/participation structure of the Witkey model, and found that, although

[15] www.mturk.com
[16] www.crowdflower.com
[17] www.taskcn.com
[18] en.wikipedia.org
[19] answers.yahoo.com
[20] www.flickr.com
[21] http://del.icio.us/
[22] reddit.com
[23] http://musicmachinery.com/tag/tagatune/
[24] images.google.com/imagelabeler/
[25] thesheepmarket.com
[26] www.youtube.com
[27] www.innocentive.com
[28] www.ninesigma.com/

it is not beneficial to the average worker, participation rates in a task increase logarithmically with the task reward.

In this chapter, we are primarily interested in crowdsourcing as an NLP annotation tool, in which human knowledge and judgment is used to determine ground truth in language tasks. Generally, this requires humans to complete small, independent tasks that use little time or expertise, using an infrastructure such as *Amazon Mechanical Turk* (MTurk) or *CrowdFlower*. In this paradigm, persons posting short webpage-based tasks (named *HITs* by MTurk) on the website are called *requesters* and persons doing the work are known as *workers*, or in the case of MTurk, *Turkers*.

2.2 Types of Crowdsourcing

Different crowdsourcing arrangements allow workers to receive various forms of compensation for their work. Different forms of compensation may be: access to website resources; entertainment value from games that collect useful metadata as a byproduct; creative design; altruism; and cash payments.

Information Collection for Access Websites sometimes need to distinguish human users from automatic or bot users. For example, many webmail services offer free email accounts to users. However, their servers would quickly be overwhelmed if bots were permitted to open new accounts. Likewise, many websites display user comments, but they need to prevent automatic postings by advertising bots in order to keep the comments interesting and appealing to humans. To identify human users, websites use a *CAPTCHA test*: an image of text is displayed, and the user must correctly enter the text of the image. The image typically displays text that is beyond current OCR capabilities, so when a user correctly enters the text of the image, they prove that they are human. To generate a fresh supply of images, *reCAPTCHA*, a variant of the text, displays two images: one with known content and the other with unknown content. The user must enter the text of both images. If the user correctly enters the text of the known image, they have proven they are human, and their answer to the second image is presumed to also be correct; the second image will then be provided as the known image to the next user. The reCAPTCHA image labeling system is a domain-specialized form of information collection crowdsourcing (Yuen et al., 2009).

Creative Design Creative tasks have also been accomplished through crowdsourcing (Yuen et al., 2011a). For example, the Sheep Market[29] (Koblin, 2009) is a web-based art project in which workers were asked to "draw a sheep facing left". 10,000 sheep drawings were collected through MTurk, and workers were paid USD$0.02 for their sheep drawing. However, as the

[29] www.thesheepmarket.com/

average time drawing a sheep was 105 seconds, and the average wage was USD$0.69/hour (which is well below standard crowdsourcing rates), workers must have agreed to participate partly due to enjoyment of creative design.

In SwarmSketch[30], a topic is provided (such as "Atlas Shrugged" or "Debt Collectors") and each user is permitted to contribute one line towards a sketch of that topic. The One Million Masterpiece project[31] asks each participant to draw the contents of one digital square, and the squares comprise a giant patchwork collaborative image; users can communicate with their neighbors and change their square's image at any time. In the Johnny Cash Project[32], users submit a drawing that will be used as one frame in a music video for Johnny Cash's song, *Ain't No Grave*, and receive name credit for their contribution.

Altruism Some crowdsourcing paradigms benefit the worker only via a sense of altruism. For example, Nam et al. (2009) interviewed 26 users of the largest South Korean question-answering site, `Naver-Knowledge-iN`, and found that top answerers cited altruism, learning, and competency as their motivations for participation. A drawback to this motivation, however, is that participation was found to be highly intermittent. Top answerers also cited the point-system and thank-you messages for increasing participation motivation.

Thousands of crowdsource volunteers looked at 560,000 satellite images in early 2007, trying to find the location of computer scientist Jim Gray, who went missing on a sailing trip that year. The effort was not successful, but did demonstrate the desire of volunteers to help with a good cause. (Quinn and Bederson, 2011)

Games With A Purpose Games with a Purpose (a.k.a. social games) are a form of online entertainment game, in which the game design allows the host to collect useful metadata generated by the game players. First described by Von Ahn (2006), the games vary in their structure to use different arrangements of data input and output, symmetric or asymmetric verification, and collaborative or competitive interaction between players (Yuen et al., 2009).

One such game is the ESP Game (Von Ahn, 2006). This game was designed to collect labels for images for accessibility applications for the visually impaired. In this game, random pairs of online players are simultaneously shown the same image. Each player tries to guess what label the other player would give the image. The more labels each player submits, the more likely there is to be a match. When a pair of players have both submitted the same label, the game moves on to the next image. Players receive points for each match, as well as a bonus for matching all 15 images in a round. The text string that the two players agreed upon is usually a good label for the image, and it is this data that is collected for the game's purpose.

[30] `http://swarmsketch.com/`

[31] `www.millionmasterpiece.com/`

[32] `http://www.thejohnnycashproject.com/`

Chan et al. (2009) propose a formal framework for the design of social games, listing out the design elements of the game, the characteristics of the human computation problem, and their relationships. They also propose a set of design guidelines derived from the formal model. Jain and Parkes (2009) discuss how game theory can be incorporated into the design of games with a purpose. For example, current game-theoretic models of social games assume that each player's interests are entirely selfish; however, a more accurate model needs to account for the altruism that crowdsource contributors are known to exhibit.

Eickhoff et al. (2012) combine the concept of games with a purpose, with paid microtask crowdsourcing, and show that the additional gaming motivation of the task permitted collection of high-quality annotations at significantly lower pay rates and with less fraudulent work.

Cash Payments Crowdsourcing arrangements motivated by cash payments, such as MTurk, CrowdFlower, and Taskcn platforms, offer the greatest task flexibility of all forms of crowdsourcing. A wide range of work can be accomplished in exchange for (even tiny) cash payments, such as image labeling, voice transcription, identification of information (such as a company name) in a document, website classification, story rating, product translation, and surveys. While information collection, creative design, and games with a purpose are uniquely tailored to convince workers to contribute data for free, they require labor-intensive task design and are not suitable for many tasks. In contrast, crowdsourcing via cash payment has a lower threshold for task design, enabling data collection for smaller tasks than are feasible with the other forms of crowdsourcing.

Quinn and Bederson (2011) also point out drawbacks of cash motivation. It may drive workers to cheat the system, and the anonymity of most workers may increase the likelihood of dishonest behavior.

While some Turkers work for fun or to kill time (51% of US workers and 62% of Indian workers in November 2009 agreed with the statement "MTurk money is irrelevant to me." or "MTurk money is nice, but doesn't materially change my circumstances." (Ross et al., 2010)), many Turkers work for the money.

2.3 A Brief History

In order to provide a context for our crowdsourcing contributions, we explain historical usage and goals of crowdsourcing. Our crowdsourcing research contributions are motivated by the recent development of crowdsourcing websites, whose cost-effective usage (Jamison and Gurevych, 2014a) and label learning (Jamison and Gurevych, 2015) is still being determined.

One of the earliest documented crowdsourcing tasks was the Longitude Prize (Wikipedia, 2015a). Prior to the 18th century, sailors could not reliably determine the east-west location of their ship when out of sight of land, and this resulted in a number of maritime disasters.

The English Parliament authorized the Longitude Act in 1714, offering a prize of up to £20,000 (£2.81 million today) to anyone who could provide a method that could determine longitude to various specificities. John Harrison, the son of a carpenter (Wikipedia, 2015b), provided the best solutions for the task, and was eventually awarded over £23,000 for his work (Wikipedia, 2015a). At the time, it was a novel concept to issue awards for work regardless of social class background.

A number of other public contests and collective-computation tasks followed, such as the 1916 competition to design a Planters Peanuts logo and the cataloging of words by 800 workers for the Oxford English Dictionary in 1884 (Wikipedia, 2015c). However, it was the development of microtask labor market websites in the 2000's, facilitating arrangements between requesters and workers and delivering tasks to workers in their own homes, that led to the explosion in use of crowdsourcing seen today. The term crowdsourcing, developed by Wired Magazine editors Jeff Howe and Mark Robinson in 2005, is a portmanteau of "outsourcing" work to the "crowd" (Wikipedia, 2015c).

The best-known crowdsourcing website, Amazon's Mechanical Turk, is named after an 18th century hoax (Wikipedia, 2015d). In 1770, Wolfgang von Kempelen, an employee of the Habsburg Court in Austria, built a cabinet with machinery and Turkish-clothed mannequin torso and chessboard, and presented the device as an automatic chess machine, "The Mechanical Turk". Although an investigation of the device appeared to show clockwork-like gears and mechanisms inside the cabinet, the cabinet secretly housed a human chess master, who monitored the game through the chessboard with magnetic game pieces and pulled levers to control the mannequin's arm movement and make the "automatic" chess moves. The device was built to impress Empress Maria Theresa of Austria, and went on to tour Europe and America for over 50 years, playing against a variety of opponents including Napoleon and Benjamin Franklin, before being exposed as a hoax in the 1820's.

The Mechanical Turk personifies the essence of crowdsourcing: chess, like modern tasks such as image labeling and linguistic annotation, is difficult to automate but easy for humans, and can be accomplished in a machine-like manner by an anonymous human hidden behind an interface.

The Amazon Mechanical Turk platform was launched in November 2005 (Ross et al., 2010). Jeff Bezos, chief executive of Amazon.com, originally created an internal crowdsourcing site to identify duplicate pages for Amazon products. The system worked well, and Bezos launched Mechanical Turk in November 2005 (Ross et al., 2010; Pontin, 2007) so that everyone could use crowdsourcing. By March 2007, Amazon stated that MTurk had over 100,000 workers from 100 countries (Pontin, 2007).

CrowdFlower, an alternative crowdsourcing website, was launched in 2007. In addition to serving as a labor market, CrowdFlower provides tools to enable the requester to clean up messy and incomplete data (Wikipedia, 2015e).

Some other crowdsource websites include CloudCrowd.com (founded before June 2010), RapidWorkers.com, Samasource.org (a non-profit organization to enable impoverished women and youth from developing countries to find work as crowdsource workers), Microworkers.com, and Clickworker.com.

2.4 Demographics

In order to understand crowdsourcing as an annotation tool, it is necessary to understand its demographic-based biases, especially in comparison with the alternative, expert labeling by university students. We provide a summary of crowdsourcing demographics below.

When Amazon launched MTurk, it offered cash payments only to workers with a US bank account; other workers were paid with Amazon gift cards. This persisted until circa 2009, when Indian bank accounts became accepted. As a result, 70-80% of early workers were American; they were representative of US internet users (Ipeirotis, 2009).

Ross et al. (2010) surveyed Turkers between March 2008 and November 2009, and found a steady increase in international Turkers, from 8% to 36%, and 46% in February 2010 (Silberman et al., 2010). Ipeirotis (2010) confirmed this, finding in February 2010 that 47% of Turkers were American and 34% were Indian. However, the Indian Turkers do not earn as much as US Turkers ($1.58/hr versus $2.30/hr), because requesters of the highest paying tasks often require a US IP address as proof of English fluency. During this time period, the gender balance shifted from 58% female to 52% female, the percentage of Turkers with household income below $10,000 rose from 22% to 32% across the year 2009, and the average age dropped from 32.9 in November 2008 to 31.6 in November 2009, which all reflected the shift towards Indian Turkers. Additionally, the study found that 27% of Indian Turkers and 14% of US Turkers use MTurk "to make basic ends meet", and only 18% of workers spend more than 15 hours per week on HITs.

Ipeirotis (2010) found that the majority (65%) of US workers were female. Turkers tend to be stay-at-home parents, unemployed and underemployed workers, and use MTurk as a supplementary course of income, and females are over-represented in these categories. Turkers in both India and the US are younger than general internet users, and have more education than the general respective populations. In contrast, 70% of Indian workers are male.

Paolacci et al. (2010) confirmed that most (65%) of US workers were female, and found that US workers have higher education but lower household income than US median; they also note that this demographic pool is much more similar to the US population in general than traditional university subject pools, and that the MTurk non-response error rate is lower than for other internet-based participant pools. Over 50% of Indian Turkers have a Bachelors degree, and 25% have a Masters degree. About 35% of American Turkers have a Bachelors degree, 15% have a Masters degree, and about 5% have a PhD. (Results are self-reported.)

While Paolacci et al. (2010) found that US Turkers have slightly lower incomes than general US population (45% of the US has a household income below $60K/yr, versus 66.7% of US Turkers), it is worth noting that 33.3% of US Turkers have a household income *above* $60K/yr, which indicates that many US workers are not dependent on their MTurk wages. About 55% of Indian Turkers have household incomes below $10K/yr. As is typical for their age demographic, most American and Indian Turkers have no children, and many are single.

Paolacci et al. (2010) found that US and Indian Turkers participated in MTurk with similar frequency: most Turkers spent less than 8 hours per week on HITs and completed around 20–100 HITs per week, which corresponds to less than $20/week income. About half of US and Indian Turkers earned less than $5/week from MTurk. About 60% and 70% of Indian and US Turkers, respectively, agreed with the statement, "Mechanical Turk is a fruitful way to spend free time and get some cash (e.g., instead of watching TV)". About 27% of Indian Turkers and 13% of US Turkers reported, "Mechanical Turk is my primary source of income (paying bills, gas, groceries, etc)."

Goodman et al. (2013) surveyed Turkers from a behavioral research perspective, and in a comparison with traditional community and student samples, found that Turkers are less likely to pay attention to research materials, reducing the statistical power of research from crowdsource origins. Turkers are more likely to use the internet to find answers, and have similar attitudes to students concerning money. They also found Turkers are less extroverted and have lower self-esteem than other participants. Otherwise, they found Turkers to react in similar behavioral ways: present biased, risk-adverse for gains, and show the certainty effect, among other traits.

Berinsky et al. (2012) found that (US) Turkers are often more representative of the US population than in-person convenience samples, the typical participant source for political science research. Buhrmester et al. (2011) found that Turkers were slightly more demographically diverse than internet samples, and much more diverse than typical American college sample, the typical source of social science data.

As of 2010, Amazon reported it had 400,000 workers registered, with 50,000–100,000 HITs (tasks to work on) available (Ross et al., 2010). However, Stewart et al. (2015) calculate, using capture-recapture analysis, that in early 2015, the average laboratory was accessing about 7,300 Turkers. Stewart et al. (2015) also calculate that it takes about 7 months for half of that population to be replaced. This means that the hundreds of laboratories using MTurk for experiment pools may be using overlapping pools of workers, or re-using the same workers for multiple experiments. This is concerning, because Chandler et al. (2015) found that effect sizes in experimental studies are reduced when the participants have previously participated in a similar experimental paradigm.

CrowdFlower, another crowdsourcing website, promotes itself as the "world's largest network of on-demand contributors", with over 5 million workers (Zukoff, 2014). Zukoff (2014) found CrowdFlower's workers come from the US (18%), India (12%), Great Britain (6%), and

151 other countries. Similar to the high education and gender-divide findings in MTurk, 25% of CrowdFlower workers have a Bachelors degree, 11% have a Masters degree, and 72% are female. 42% are white and 30% are Asian. The majority of workers are younger than 30, have never married, and have no children. 81% of workers own a smartphone. 46% of worker have a household income below USD$10,000/year. The most frequent reason for worker participation (50%) was, "it is a great way to spend free time and get some cash"; only 7% cited crowdsource work as their primary source of income.

Crowdsourcing as a labor marketplace has shown that internet-based human labor is very cheap. Crowdsource workers, even from wealthy, high-living-cost countries such as the US, are generally available at USD$2 or $3 per hour (Ross et al., 2010); meanwhile, US federal minimum wage is currently USD$7.25/hr, and some areas have set minimum wage at $15/hr. This has led to suggestions that crowdsourcing is a form of exploited labor (Zittrain, 2009).

However, a study by Horton (2011) showed that workers perceive online employers as slightly fairer and more honest than offline employers (although the effect is not significant). Horton (2011) also points out that crowdsourcing markets give people in poor countries access to buyers in rich countries, an enormous benefit. Additionally, compared to other forms of labor in poor countries, crowdsourcing involves no physical danger to workers, causes no environmental degradation, and does not expose workers to required long hours, unpredictable agriculture, or tyrannical bosses.

Furthermore, Mason and Watts (2009) disproved the concept that raising wages increased the quality of crowdsourced work: they found that workers who were paid more also felt the value of their work was greater, resulting in no increase motivation over lower-paid workers.

2.5 Annotation Tasks

Crowdsourcing has been used for a wide variety of tasks. Almendra and Schwabe (2009) applied crowdsourcing to the task of identifying fraud among online auction site sellers, and showed that Turkers were effective at identifying the fraudulent sellers based just on the seller profile, before the arrival of negative transaction feedback. Holmes et al. (2009) use crowdsourcing to identify bacteria. Sorokin and Forsyth (2008) investigate two methods to efficiently obtain crowdsource annotations for a computer vision task.

In NLP, Snow et al. (2008) used Turkers to annotate data for five tasks: numerical judgments in the interval [0,100] for six emotions in news headlines in an affective text task, word similarity judgments for word pairs using a scale of [0,10], binary textual entailment judgments for pairs of sentences where one sentence might be inferred from the other, temporal ordering of pairs of verbs from an event description, and sense judgments in a word sense disambiguation task. They found high agreement between Turker judgments and pre-existing expert annotators, and showed that aggregated Turker judgments could often outperform the experts.

Task design is critical to effective crowdsourcing. A successful task provides brief and unambiguous task instructions. Larger tasks may need to be broken down into small components that can be crowdsourced. Kittur and Kraut (2008) shed light on the dynamics of labor division with their research on Wikipedia article quality and increasing numbers of editors contributing to the article: articles did not automatically improve in quality with more editors or with explicit discussion between editors, but only if the editors were implicitly managed by a small group of editors who do the majority of the work and set direction for everyone else.

Kittur et al. (2011) present the framework CrowdForge, which accomplishes large tasks by assisting the requester to break down the task into small interdependent components that can be completed in the microtask crowdsource marketplace. Little et al. (2009) present TurKit, a toolkit for running a large task as a single program that uses MTurk workers as subroutines for small interdependent tasks within the large task.

2.6 Economic Issues

MTurk's platform design allows requesters to reject work without paying, based on perceived low quality, without publishing a record of the requester's history of rejecting work, which puts workers at risk for wage theft. Silberman et al. (2010) presents TurkOpticon[33], a worker-side Firefox add-on that augments MTurk's list of available HITs by adding worker-written reviews of requesters.

Crowdsourcing is more effective and efficient when workers are well-matched to tasks. In crowdsourcing, workers choose their own tasks; however, a worker generally only browses a few pages of tasks before choosing one to complete. Chilton et al. (2010) found that tasks listed high on the task display were completed 30 times faster and for less money than low-listed tasks. Yuen et al. (2011b) propose an algorithm to better match workers with tasks that they chose previously and performed well, by displaying such tasks high in a task search retrieval. A user study with 40 tasks and 10 participants showed that task matching was more accurate in predicting worker task preferences than a random baseline.

In order to plan their HITs to match real-world data needs, it is necessary for requester to be able to estimate how long it will take to complete a task. Some tasks are completed in a matter of minutes, while others can drag on for weeks. Wang et al. (2011b) analyze 6.7 million completed HITs from 165,000 HIT groups. They model the completion time as a stochastic process, and present a time-to-completion algorithm based on Cox proportional hazards regression.

A requester can more effectively crowdsource their work when they can calculate a worker's *reservation wage*, the smallest wage a worker is willing to accept to complete a task. (Incidentally, this median wage on MTurk was USD$1.38/hr in 2010.) Horton and Chilton

[33] `turkopticon.differenceengines.com`

(2010) present a model to calculate reservation wage. They also discuss other factors, such as financial incentives and earning targets, that influence worker behavior. For example, certain workers prefer earning total amounts evenly divisible by 5, presumably because these numbers make good targets. Moreno et al. (2009) calculated that, for a question-answering website, answering participation is greatest when the site offers both long-term and short-term rewards for the workers.

2.7 Label Quality

Multiple studies have shown that crowdsourced annotation is equivalent in quality[34] to expert annotation. Nowak and Rüger (2010) compare expert annotations with crowdsourced annotations from an image labeling task. While agreement between experts and between crowdsource workers varied based on the agreement measure, its impact on systems is small, and aggregating crowdsource labels with majority vote was effective at filtering some label noise. Snow et al. (2008) investigate the inter-annotator agreement on five natural language tasks, and find very high agreement between Turkers and expert annotators. For an affect recognition task, they show that equally-performing machine classifiers can be trained using crowdsource or expert labels. Similar conclusions on crowdsource label quality have been reached by Zaidan and Callison-Burch (2011), Gao and Vogel (2010), Sprouse (2011), and others.

Nevertheless, identifying and addressing specific quality problems in crowdsourced annotation results in better annotations. Several types of problems can reduce the quality of annotations obtained from crowdsource workers, including spam and worker fraud, mistakes, low worker quality, and worker bias. Although all of these problems reduce label quality, they impact the data in specific and differentiated ways, and must be controlled with different techniques.

Types of crowdsource label quality problems:

- *Spam* and *worker fraud* happens when crowdsource work is submitted without a human doing any work. This may result from a bot crawling a crowdsource website and submitting work with random answers, in the hope that the requester will send payment without screening for work quality. It may also result from a human submission where the human randomly clicked checkboxes or pasted text without reading the instruction, or clicked the first answer for every task.

[34] Annotation quality is traditionally quantified via an inter-annotator agreement measurement. However, Passonneau and Carpenter (2014) demonstrate that this is ineffective, showing that low agreement does not necessarily indicate label uncertainty, and high agreement does not guarantee label certainty, and that instead, aggregate annotation quality should be judged via a Dawid-and-Skene-style 1979a statistical model that calculates individual instance certainty after estimating annotator quality and bias. Until more results are available showing the impact of statistical modeling for measuring annotation quality on NLP research, we will follow the bulk of the literature and assume a high IAA measure denotes label certainty and corpus quality.

	% Affected labels from the worker	*Do affected labels contain information?*	*With a proper cofactor, is gold anno calculable for a single instance?*
spam/fraud	all	no	no
mistakes	some	no	no
worker quality	all	yes	no
worker bias	all	yes	yes
instance ambiguity	some	yes	no

Table 2.1: Quick comparison of crowdsource worker-triggered label problems.

- *Mistakes* are accidental mis-labels from an otherwise competent worker.
- *Worker quality* describes that workers vary systematically in the quality of their work. Some workers routinely submit work that is highly correlated with the work of other workers, while other workers routinely submit outlier work. For some tasks, this may be due to differences in worker expertise.
- *Worker bias* reflects the different mental thresholds at which workers decide between labels. After worker bias has been taken into account, a previously-noisy-appearing labelset may be found to have perfect inter-annotator agreement. An extreme case is a contrary or argumentative worker who, in a binary classification annotation, always submits the opposite label from everyone else.
- *Ambiguous instances* in the dataset will cause low inter-annotator agreement even when there is no spam/fraud, mistakes, low worker quality, or bias. Often, it is only possible to identify ambiguity after ruling out all the other possible forms of worker error.

Table 2.1 provides a quick-guide comparison between the different types of worker-triggered label problems and instance ambiguity. As can be seen in the table, the affected percentage of labels varies by the type of problem, as does the amount of information in an affected label and the ability to reconstruct the gold label for a single instance given a cofactor.

2.7.1 Spam and Worker Fraud

Kittur et al. (2008) provide one of the earliest descriptions of "gaming the system", on a task asking workers to read a Wikipedia article, assign a rating, and suggest needed improvements in a free-form text box:

> Out of the total of 210 free-text responses regarding how the article could be improved, 102 (48.6%) consisted of uninformative responses including semantically empty (e.g., "None"), non-constructive (e.g., "well written"), or copy-and-paste responses (e.g., "More pictures to break up the text" given for all articles rated by a user). An examination of the time taken to complete each rating also suggested gaming, with 64 ratings completed in less than 1 minute (less time than likely needed for reading the article, let alone rating it). 123 (58.6%) ratings were flagged as potentially invalid based either on their comments or duration. (Kittur et al., 2008)

Rashtchian et al. (2010) note that MTurk tasks collecting free text are particularly difficult for screening work, because there are multiple correct answers, so control items cannot be embedded in the task. They evaluate several quality control strategies, and find that using a qualification test provides the biggest boost in quality, while refining annotations in follow-up tasks works poorly.

Silberman et al. (2010) points out that requesters may be gaming the system just as much as workers, by: posting tasks in broken HTML that collect work but never get submitted to pay workers; or posting completely broken hits that waste workers' time but reject the work because it is impossible to submit quality work on a broken task; or requesting workers to perform illegitimate tasks like filling CAPTCHAs, secret shopping, testing webpages, clicking links, sending text messages, or submitting personal information.

Quinn and Bederson (2011) suggest a variety of tactic to combat cheating in crowdsourcing, including defensive task design, a reputation system, redundancy, ground truth seeding, statistical filtering, and multilevel review. Defensive task design describes designing a task so that it is no easier to cheat than it is to actually do the work. A reputation system keeps track of a worker's history, so that workers with low performance or bad work history can be blocked from future tasks. Using redundancy sends a single task to multiple workers and combines or aggregates their results via voting, under the assumption that most of the workers will provide good answers and outvote the outliers. (This assumption may not always be true, especially when workers know that their work will be judged via agreement with other workers (Martinez Alonso, 2013). And Sheng et al. (2008) prove that redundancy may or may not be effective depending on the quality of the individual labels.) One form of this, majority voting, is formalized by Hirth et al. (2010).

Ground truth seeding is the mixing of questions with known answers among the questions that need to be solved by the Turkers. A worker that is deliberately submitting bad answers can be identified by failing the seeded questions. Statistical filtering refers to techniques of filtering the data to remove outlier results. In multi-level review, the task is completed in two rounds: the first round of workers does the work, while the second round of workers checks the work quality. Hirth et al. (2010) points out that this is specifically effective for tasks that are much cheaper to confirm than they are to initially complete. Much crowdsourced work

that is described in peer-reviewed papers, including the work in this thesis, combines these techniques.

Downs et al. (2010) propose a specific qualifying test designed to catch workers without good intentions. Their test is not specific to the task at hand, but is a simple reading comprehension task with an easy and a hard question. A demographic analysis shows that young men are most likely to fail at this qualification task, while professionals, students, and non-workers are most likely to pass.

Eickhoff and de Vries (2011) investigate the behavior of malicious workers, and determine that such workers are less frequently encountered in novel tasks that require creativity and abstraction. They also observe that while pre-task filtering by worker origin can also significantly reduce the number of malicious workers, such tactics are less preferable than implicit crowd filtering by task design. They suggest that if worker reputation is to be used as a filter, then a more sophisticated system should be developed, recording information about a worker such as refusal to complete free text fields in favor of check boxes, etc.

Buchholz and Latorre (2011) investigate crowdsourcing judgments on a speech synthesis task indicating preference on two audio clips. The researchers point out that any worker who does not play both audio clips must be cheating. They collect a worker pool of these known cheaters, and model their annotations, and use this model to identify cheaters from among workers who did play both audio clips.

Tarasov et al. (2014) propose a multi-armed-bandits approach for real-time annotator quality assessment. This approach is particularly well-suited for crowdsourcing, because it does not require each worker to be available to give a label at the time as such label is identified as being needed, unlike alternatives such as the active learning approach in Wu et al. (2013).

2.7.2 Mistakes

All annotators sometimes make mistakes, due to distraction, etc. Among a pool of high quality workers, *mistakes* are easily detectable with redundant labeling and label aggregation, as described in Section 2.7.1. Sheng et al. (2008) investigate the impact of redundant labeling in more detail, showing that redundant labeling is preferable to single labeling in the face of label noise, even when labels are not cheap, and that when the cost of processing unlabeled data is not free, redundant labeling is highly advantageous. They also investigate selection of data points which most benefit from redundant labeling, and show that the redundant labeling of carefully-selected data yields the best results.

Annotators can be trained, and annotation quality maintained, through the use of hidden gold instances; the known correct label of these instances allows researchers to quickly estimate the performance of an annotator. Oleson et al. (2011) show that such instances can be altered to resemble particular types of frequent annotation mistakes, such as the confusion of two businesses with the same name in a URL verification task. These altered gold instances,

known as pyrite, can be used to enhance annotator training by forcing annotators to encounter highly mistakable instances, where they receive instant correction and auto-generated feedback.

Dligach and Palmer (2011) propose two different error detection algorithms, with a special aim at reducing the need of redundant annotation. In the first algorithm, a machine classifier learns a model on part of the dataset, and makes predictions for the other part; incorrectly classified instances are more likely to have mistaken labels. In the second algorithm, inspired by uncertainty sampling (a form of active learning), a machine classifier is trained on part of the dataset, and high-uncertainty instances are likely to have mistake labels, and are selected for redundant annotation.

In a relation extraction task, Min and Grishman (2012) developed a technique for identifying errors specific to a relation extraction task in the ACE 2005 corpus that focuses on improving negative-class precision and applying transductive inference to utilize unextracted instances during the training phase. The algorithm learns from cheap single-pass annotations and produces performance similar to the more expensive three-pass redundant annotation.

When worker quality assessment does not distinguish between bias and error (mistakes), biased workers who put a lot of thought into their labels may be unfairly penalized. Ipeirotis et al. (2010) present a Bayesian EM modeling technique that separately models bias and error by using expected cost of a soft label to model worker quality. Welinder and Perona (2010) present a Bayesian EM algorithm that separately models label uncertainty (worker mistakes) and worker quality, allowing derivation of integrated labels with a desired level of confidence, and allowing exclusion of unreliable workers. This algorithm is also unique in its ability to handle a wide variety of annotation types, including binary, multi-valued, and continuous annotations. Experiments show that this method reduces the number of labels required while keeping error rates low.

2.7.3 Worker Quality

Sometimes, low *worker quality* may be the result of inattention when reading the task instructions. An Instructional Manipulation Check (IMC) is a mini-quiz to make sure the worker read the instructions. Hauser and Schwarz (2015) describes one such IMC:

> Under the header "Sports Participation" respondents were asked, "Which of these activities do you engage in regularly? (click on all that apply)." However, above the question, a block of instructions (in smaller text) indicated that, to demonstrate attention, respondents should click the other option and enter "I read the instructions" in the corresponding text box. Following these instructions was scored as a correct response to the IMC.

Hauser and Schwarz (2015) found in a MTurk study that 92% of workers correctly answered this IMC. However, they also found that the presence of the IMC changed the wor-

kers' responses to the actual task of math problems, suggesting that experimenters should be cautious in their use of IMC's and interpretation of their experimental results.

Low work quality might also be the result of inattention during the task itself. Researchers sometimes include an attention check question (ACQ) in the middle of the task, such as "Have you ever, while watching TV, had a fatal heart attack?" (Paolacci et al., 2010). However, Peer et al. (2014) found that the use of ACQ's did not improve work quality among high-reputation workers. They also found that, although ACQ's increase work quality among low-productivity workers, this work quality was inferior to the work quality of high-productivity workers, suggesting that ACQ's are not necessary and worker reputation and productivity is sufficient.

When annotators each label many examples and examples are redundantly labeled, a ground truth can be calculated from the majority vote for each instance, and low-performing annotators identified by counting how often an annotator's labels match the majority vote. However, in many crowdsource tasks, each worker only labels a few instances, and the number of workers scales with the size of the task. In this situation, to identify low quality workers, Dekel and Shamir (2009) propose to train a hypothesis on the entire unfiltered dataset, and treat the predictions as approximate ground-truth; then, worker quality can be judged by agreement with the predictions, and work from low quality annotators pruned away. An evaluation showed this technique reduced noise in the dataset even when there were too few labels per annotator to reliably estimate each annotator's quality. Raykar et al. (2010b) uses a similar but iterative technique: a Bayesian Expectation Maximization (EM) algorithm iteratively discovers ground-truth labels via logistic regression machine learning on the dataset, measures the performance of the annotators against the newly-discovered ground truth labels, re-weights the quality of the annotators (and their labels) accordingly, and repeats.

Latent Bayesian modeling was first proposed by Dawid and Skene (1979a), who iteratively and simultaneously determined true medical diagnoses, as well as the skill level of the doctors, using judgments of multiple doctors for each case.

Methods of redundant label aggregation that account for individual worker quality, such as Dawid and Skene's (1979b) latent Bayesian modeling, perform better when all annotators annotate all instances. However, in crowdsourcing, many workers may only label a few instances each. Jung and Lease (2012) propose using probabilistic matrix factorization (PMF) to estimate the "missing" labels of the sparse label matrix, and show that PMF with majority vote aggregation matched the performance of the Bayesian modeling, on both a supervised and an unsupervised task.

2.7.4 Worker Bias

Worker bias is quality error that can be algorithmically corrected specific to each individual task. For example when answering a survey where the participant must rate each item on a scale of 1-5, there will be some participants who rate all items as 4's and 5's, and some

participants who rate all items as 1's and 2's; a "5" from the former type of participant is equivalent to a "2" from the latter type of participant.

Carterette and Soboroff (2010) propose human-realistic models for seven types of worker bias, as supported by examination of information retrieval (IR) assessment labels; although derived from an IR task, the models are relevant for many multiple-choice crowdsource annotation tasks.

The *unenthusiastic* worker is not interested in the content of the texts and simply wants to finish the job and be paid; the associated work pattern is to judge all texts negatively, or alternating judgments in some pattern; completion time is faster than normal. The author of this thesis has also observed this behavior among Turkers who always select the first option in a multiple choice list. This model is the most possibly biased model; the work contains no information, and the work pattern is equivalent to fraud (Section 2.7.1).

The *optimistic* worker labels most documents positively, even if there is only a small amount of evidence supporting this conclusion. For the same reasons, the *pessimistic* worker labels most documents negatively.

The *topic-disgruntled* worker chose this task due to interest in the topic or it looked easy, etc., but became disenchanted when expectations were not borne out. After the first few judgments, the worker becomes disgruntled and begins to click through more rapidly, and rates remaining documents negatively. The *lazy/overfitting* worker has similar behavior due to a different motivation: if the first n documents are all positive or all negative, the worker incorrectly assumes that the rest will be the same way, and begins to enter rapid judgments accordingly.

The *fatigued* worker starts the first task alert and attentive, but gets tired over time, and their judgments become more random.

The final model of worker bias is the *Markovian* model, where a worker's judgments are conditional on their past judgments; perhaps the worker feels they made too many of one type of judgment in the past, and they try harder to make opposite judgments on future documents.

Feng et al. (2010) found that the information provided to Turkers in an information extraction (IE) task changed the bias of the workers, impacting inter-annotator agreement rates. Wauthier and Jordan (2011) point out that modeling the effects of labeler bias by assuming a single true latent label is inappropriate for subjective or ambiguous tasks. They propose a bias mitigation system for crowdsourcing, using a Bayesian model with flexible latent features, to model labelers as influenced by shared random effects. They also show that the model is compatible with active learning.

2.7.5 Ambiguity

Finally, low inter-annotator agreement can also be caused by *instance ambiguity*, also known as *Hard Cases* or difficult cases (Beigman Klebanov and Beigman, 2014). Beigman and Kle-

banov (2009) suggest that one of the most difficult properties of ambiguous instances is that they are far more vulnerable to influence from different annotators' biases and preferences. While a worker bias model, such as those described in Section 2.7.4, might be a general estimate for the entire corpus, it does not accurately model the severe biases applied to Hard Cases.

Beigman Klebanov and Beigman (2014) investigate Hard Cases in the annotation and classification task of classifying words in a text as semantically old or new. They found evidence that the presence of Hard Cases in the training data misleads the machine learner on easy, clear-cut cases.

In a dependency parsing task, Schwartz et al. (2011) show that linguistically ambiguous instances can significantly alter unsupervised parser performance, and provide a new evaluation measure that reduces the impact of these ambiguous instances in evaluation.

Reidsma and Carletta (2008) show that, while classifiers can handle noisy annotations if that noise is random, systematic annotation disagreement (resulting from ambiguity) can introduce patterns that make evaluation results look better than they really are, suggesting that low inter-annotator agreement in a training dataset that was caused by ambiguity may produce an inferior classifier model.

2.8 Chapter Summary

In this chapter, we introduced crowdsourcing as labor market, and more specifically, as an annotation method. We discuss different forms of crowdsourcing and various motivations such as creativity, altruism, game enjoyment, and cash payments. We provide a brief history of crowdsourcing, showing that the earliest documented tasks stretch back to the 18th century, but that the modern crowdsourcing boom is due to the development of microtask websites in the 2000's. We discuss the demographics of the crowdsource workforce. We describe some tasks that have been successfully completed via crowdsourcing. We discuss economic issues of the crowdsource labor market. We describe problems with crowdsource annotation quality, including fraud/spam, mistakes, low worker quality, worker bias, and ambiguous instances, and we discuss techniques to overcome these problems.

For the rest of Part I of this thesis, we apply insights from this chapter to specific problems encountered while building a crowdsourced thread reconstruction dataset. In Chapter 3, we investigate the cost problem of crowdsource-annotating a heavily class-imbalanced dataset. In Chapter 4, we compare different techniques to learn classifier models from redundantly-labeled crowdsource datasets.

Chapter 3

Crowdsourcing Annotation of Class-Imbalanced Datasets

In order to study automatic discussion thread reconstruction, it is necessary to have a discussions corpus. Because the final desired discussion structure is a directed graph, where nodes are discussion turns and edges are the reply-to relations between the turns, it is necessary that the edges in the discussions corpus are labeled with gold-standard relations. This means that, in practical terms, a human annotator must read many pairs of discussion turns, and label most of the turns as *negative* (e.g., in adjacency recognition, not reply-to) and label a few of the turns as *positive* (e.g., reply-to). Such a *class-imbalanced* dataset is expensive to annotate, because so much unknown-class data must be labeled in order to find a few positive instances.

In the previous chapter, we have introduced the technique of crowdsource annotation and have described different forms of crowdsourcing, as well as history, demographics, tasks, economic issues, and label quality problems. We have shown how crowdsourcing has drastically reduced the cost of many annotation projects, enabling the creation of a wide variety of new datasets for empirical study of previously-unanalyzed natural language tasks. We have also shown that crowdsource annotation is noisier than trained annotation, and much previous research has worked to maximize crowdsource annotation quality and reduce the cost associated with this noise.

In this chapter, we discuss our approaches to class-imbalanced annotation, and how annotation costs may be reduced. By developing techniques to detect *common-class instances* (instances from the class with high prior probability) and therefore reduce the cost of class-imbalanced annotation, we can enable the annotation of thread reconstruction corpora, which are needed for investigations like those described in Chapter 6, Chapter 7, and Chapter 8. To learn more about class-imbalanced annotation and crowdsourcing, we address the following research questions:

Research Question: It has been shown that annotation quality on a class-balanced dataset is improved by redundant labeling. Should a class-imbalanced dataset be redundantly crowd-labeled?

Research Question: How cost effective is discarding instances that receive a single common-class label, compared to a trained, metadata-feature-based classifier cascade, to acquire *rare-class* (i.e., instances from the class with low prior probability) instances?

The chapter is structured as follows. First, we provide an overview of our motivation (Section 3.1), and a discussion of previous research (Section 3.2). We investigate the class imbalance of three different crowdsource annotation tasks (Section 3.3) and their baseline costs (Section 3.4). We describe our experiments with supervised cascading classifiers (Section 3.5) and our experiments with rule-based cascades (Section 3.6). We conclude the chapter with a summary of our findings (Section 3.7).

Most of the material in this chapter was previously published in peer-reviewed proceedings:

Emily K. Jamison and Iryna Gurevych: 'Needle in a Haystack: Reducing the Costs of Annotating Rare-Class Instances in Imbalanced Datasets', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing* (PACLIC), Phuket, Thailand, 2014.

3.1 Motivation

The advent of crowdsourcing as a cheap but noisy source for annotation labels has spurred the development of algorithms to maximize quality and while maintaining low cost. Techniques can detect spammers (Oleson et al., 2011; Downs et al., 2010; Buchholz and Latorre, 2011), model *worker quality* and *bias* during label aggregation (Jung and Lease, 2012; Ipeirotis et al., 2010) and optimize the decision to obtain more labels per instance or more labeled instances (Kumar and Lease, 2011; Sheng et al., 2008). However, much previous work for quality maximization and cost limitation assumes that the dataset to be annotated is class-balanced.

Class-imbalanced datasets, or datasets with differences in prior class probabilities, present a unique problem during corpus production: how to include enough rare-class instances in the corpus to enable machine classification? If the original class distribution is maintained, a corpus that is just large enough for a classifier to learn *common-class* (i.e., frequent class) instances may suffer from a lack of *rare-class* (i.e., infrequent class) instances. Yet, it can be cost-prohibitive to expand the corpus until enough rare-class instances are included.

Content-based instance targeting can be used to select instances with a high probability of being rare-class. For example, in a binary class annotation task identifying pairs of emails from the same thread, where most instances are negative, cosine text similarity between the

emails can be used to identify pairs of emails that are likely to be positive, so that they could be annotated and included in the resulting class-balanced corpus (Jamison and Gurevych, 2013). However, this technique renders the corpus useless for experiments including token similarity (or n-gram similarity, semantic similarity, stopword distribution similarity, keyword similarity, etc) as a feature; a machine learner would be likely to learn the very same features for classification that were used to identify the rare-class instances during corpus construction. Even worse, Mikros and Argiri (2007) showed that many features besides n-grams are significantly correlated with topic, including sentence and token length, readability measures, and word length distributions. The proposed targeted-instance corpus is unfit for experiments using sentence length similarity features, token length similarity features, etc.

Active learning presents a similar problem of artificially limiting rare-class variety, by only identifying other potential rare-class instances for annotation that are very similar to the rare-class instances in the seed dataset. Rare-class instances may never be selected for labeling if they are very different from those in the seed dataset.

In this chapter, we explore the use of cascading machine learner and cascading rule-based techniques for rare-class instance identification during corpus production. We avoid the use of content-based targeting, to maintain rare-class diversity, and instead focus on crowdsourcing practices and metadata. To the best of our knowledge, our work is the first work to evaluate cost-effective non-content-based annotation procedures for class-imbalanced datasets. Based on experiments with three class-imbalanced corpora, we show that redundancy for noise reduction is very expensive on a class-imbalanced dataset, and should be discarded for instances receiving a single common-class label. We also show that this simple technique produces annotations at approximately the same cost of a metadata-trained machine classifier, or about 70% cheaper than 5-vote majority-vote aggregation, and requires no training data, making it suitable for seed dataset production.

3.2 Previous Work

The rise of crowdsourcing has introduced promising new annotation strategies for corpus development.

Crowdsourced labels are extremely cheap. In a task where workers gave judgments rating a news headline for various emotions, Snow et al. (2008) collected 7000 judgments for a total of US$2. In a computer vision image labeling task, Sorokin and Forsyth (2008) collected 3861 labels for US$59; access to equivalent data from the annotation service `ImageParsing.com`, with an existing annotated dataset of 49,357 images, would have cost at least US$1000, or US$5000 for custom annotations.

Crowdsourced labels are also of usable quality. On a behavioral testing experiment of tool-use identification, Casler et al. (2013) compared the performance of crowdsource workers, social media-recruited workers, and in-person trained workers, and found that test results among

the 3 groups were almost indistinguishable. Sprouse (2011) collected syntactic acceptability judgments from 176 trained undergraduate annotators and 176 crowdsource annotators, and after removing outlier work and ineligible workers, found no difference in statistical power or judgment distribution between the two groups. Nowak and Rüger (2010) compared annotations from experts and from crowdsource workers on an image labeling task, and they found that a single annotation set consisting of majority-vote aggregation of non-expert labels is comparable in quality to the expert annotation set. Snow et al. (2008) compared labels from trained annotators and crowdsource workers on five linguistic annotation tasks. They created an aggregated *meta-labeler* by averaging the labels of subsets of n non-expert labels. Inter-annotator agreement between the non-expert meta-labeler and the expert labels ranged from .897 to 1.0 with n=10 on four of the tasks.

Sheng et al. (2008) showed that although a machine learner can learn from noisy labels, the number of needed instances is greatly reduced, and the quality of the annotation improved, with higher quality labels. To this end, much research aims to increase annotation quality while maintaining cost.

Annotation quality can be improved by removing unconscientious workers from the task. Oleson et al. (2011) screened spammers and provided worker training by embedding auto-selected *gold instances* (instances with high confidence labels) into the annotation task. Downs et al. (2010) identified 39% of unconscientious workers with a simple two-question qualifying task. Buchholz and Latorre (2011) examined cheating techniques associated with speech synthesis judgments, including workers who do not play the recordings, and found that cheating becomes more prevalent over time, if unchecked. They examined the statistical profile of cheaters and developed exclusion metrics.

Separate weighting of worker quality and bias during the aggregation of labels can produce higher quality annotations. Jung and Lease (2012) learned a worker's annotation quality from the sparse single-worker labels typical of a crowdsourcing annotation task, for improved weighting during label aggregation. In an image labeling task, Welinder and Perona (2010) estimated label uncertainty and worker ability, and derived an algorithm that seeks further labels from high quality annotators and controls the number of labels per item to achieve a desired level of confidence, with fewer total labels. Tarasov et al. (2014) dynamically estimated annotator reliability with regression using multi-armed bandits, in a system that is robust to annotator unavailability, no gold standard, and label type variety (such as regression, binary, and multi-class classification). Dawid and Skene (1979a) used an EM algorithm to simultaneously estimate worker bias and aggregate labels. Ipeirotis et al. (2010) separately calculated bias and error, enabling better quality assessment of a worker.

Some research explores the trade-off between obtaining more labels per instance or more labeled instances. Sheng et al. (2008) examined machine learning performance with different corpus sizes and label qualities. They showed that repeated labeling is preferable to single labeling even when labels are not cheap, and especially when the cost of processing unlabeled

data is not free, and that best results are obtained when labeling a carefully chosen set of instances. Kumar and Lease (2011) built on the model by Sheng et al. (2008), showing that the addition of annotator accuracies resulted in better and faster learning.

Other work focuses on correcting a particular instance's label, based on properties of that instance. Dligach and Palmer (2011) used annotation-error detection and ambiguity detection to identify instances in need of additional labels. Hsueh et al. (2009) modeled annotator quality and ambiguity rating to select highly informative yet unambiguous training instances.

Alternatively, class imbalance can be accommodated during machine learning, by resampling and cost-sensitive learning. Das et al. (2014) used density-based clustering to identify clusters in the instance space: if a single cluster's internal class prior imbalance exceeds a threshold, then the cluster is undersampled to increase class balance in the overall dataset. Batista et al. (2004) examined the effects of sampling for class-imbalanced reduction on 13 datasets and found that oversampling is generally more effective than undersampling. They evaluated oversampling techniques to produce the fewest additional classifier rules. Elkan (2001) proved that class balance can be changed to set different misclassification penalties, although he observed that this is ineffective with certain classifiers such as decision trees and Bayesian classifiers, so he also provided adjustment equations for use in such cases.

One option to reduce annotation costs is the classifier cascade. The Viola-Jones cascade machine learning-based framework (Viola and Jones, 2001) has been used to cheaply classify easy instances while passing along difficult instances for more costly classification. Classification of annotations can use annotation metadata: Zaidan and Callison-Burch (2011) used metadata crowdsource features to train a system to reject bad translations in a translation generation task. Cascaded classifiers are used by Bourdev and Brandt (2005) for object detection in images and Raykar et al. (2010a) to reduce the cost of obtaining expensive (in money or pain to the patient) features in a medical diagnosis setting. In this chapter, we evaluate the use of metadata-based classifier cascade, as well as rule cascades, to reduce annotation costs.

3.3 Three Crowdsource Annotation Tasks

We investigate three class-imbalanced annotation tasks; all are pairwise classification tasks that are class-imbalanced due to factorial combination of text pairs.

Pairwise Email Thread Disentanglement A pairwise email disentanglement task labels pairs of emails with whether or not the two emails come from the same email thread (a *positive* or *negative* instance). The Ecd dataset[35] consists of 34 positive and 66 negative instances[36], and simulates a server's contents in which most pairs are negative (common-class). The emails

[35] www.ukp.tu-darmstadt.de/data/text-similarity/email-disentanglement/

[36] To reduce our experiment costs, this dataset has an artificially high class balance from what could be expected on a real email server.

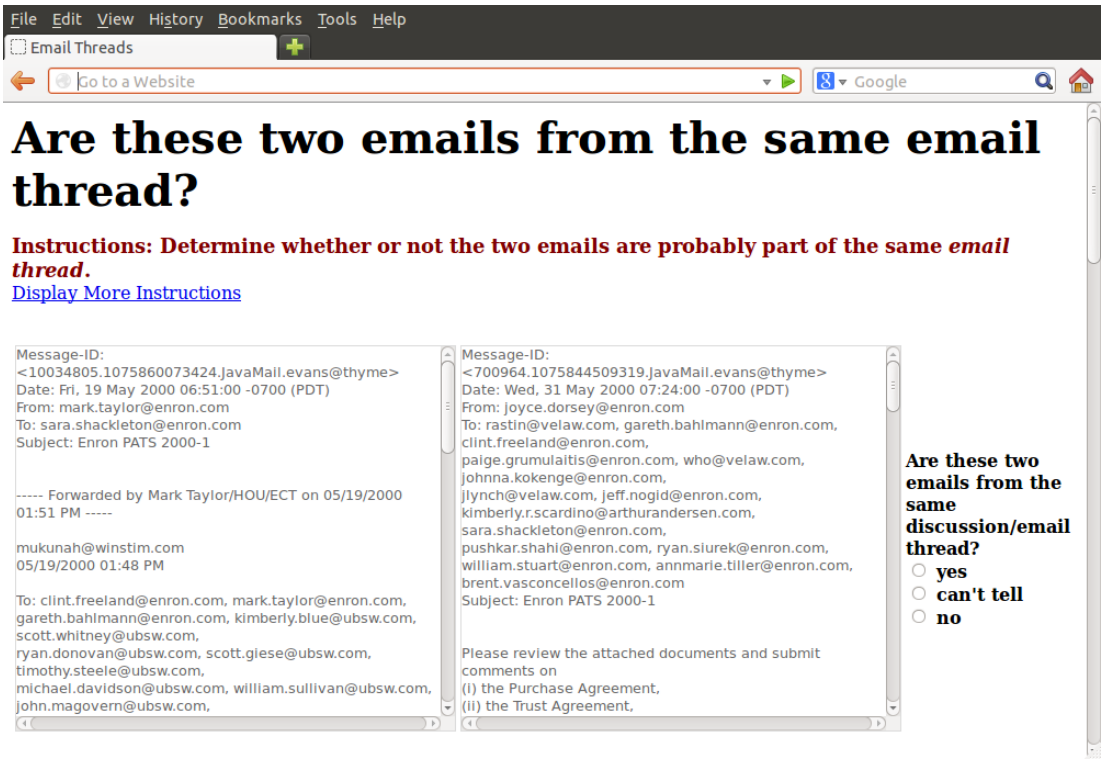

Figure 3.1: A sample MTurk HIT showing an emails pair.

come from the Enron Email Corpus , which has no inherent thread labeling. Annotators were shown both texts side-by-side and asked "Are these two emails from the same discussion/e-mail thread?" Possible answers were *yes*, *can't tell*, and *no*. Work eligibility included >94% approval rating, over 2000 HITs completed, worker location in the US. A sample emails pair is shown in the MTurk HIT in Figure 3.1.

We follow Daxenberger and Gurevych (2014) in reporting ECD inter-annotator agreement in average pairwise percentage agreement, which is calculated as $\frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{c=1}^{C} v_i^c}{C}$, where $N = 750$ is the overall number of annotated edit-turn-pairs, $C = \frac{R^2-R}{2}$ is the number of pairwise comparisons, $R = \{9 or 10\}$ is the number of raters per edit-turn-pair, and $v_i^c = 1$ if a pair of raters c labeled edit-turn-pair i equally, and 0 otherwise. The ECD average pairwise percentage agreement is 0.94, which shows strong agreement between annotators.

Pairwise Wikipedia Discussion Turn/Edit Alignment Wikipedia editors discuss plans for *edits* in an article's *discussion page*, but there is no inherent mechanism to connect specific discussion turns in the discussion to the edits they describe. A corpus of matched turn/edit pairs permits investigation of relations between turns and edits. The ETP-GOLD dataset[37] consists of 750 turn/edit pairs. Additional rare-class (positive) instances were added to the corpus,

[37] www.ukp.tu-darmstadt.de/data/discourse-analysis/wikipedia-edit-turn-pair-corpus/

Figure 3.2: A sample MTurk HIT showing a turn/edit pair.

resulting in 17% positive instances. Annotators were shown the article topic, turn and thread topic, the edit, and the edit comment, and asked, "Does the Wiki comment match the Wiki edit?" Possible answers were *yes*, *can't tell*, and *no*. Work eligibility included >96% approval rating, over 2000 HITs completed, worker location in the US, and passing a qualification exam of a sample HIT. A sample turn/edit pair is shown in the MTurk HIT in Figure 3.2. The average pairwise percentage agreement is 0.66 (Daxenberger and Gurevych, 2014).

Sentence Pair Text Similarity Ratings To rate *sentence similarity*, annotators read 2 sentences and answered the question, "How close do these sentences come to meaning the same thing?" Annotators rated text similarity of the sentences on a scale of 1 (minimum similarity) to 5 (maximum similarity). This crowdsource dataset, SentPairs, was produced by Bär et al. (2011). The Spearman correlation between the crowdsourced aggregated results and the original scores is $p = 0.91$ (Bär et al., 2011). An example sentence pair is shown in Figure 3.3. The SentPairs dataset consists of 30 sentence pairs.

The gold standard was calculated as the mean of a pair's judgments. However, on a theoretical level, it is unclear that mean, even with a deviation measure, accurately expresses annotator judgments for this task. For example, should an instance with labels [1,1,5,5,5] have the same gold standard as an instance with labels [2,2,3,3,3]? Our experiments (see Sections 3.5

Sentence1: *Cord is strong, thick string.*
Sentence2: *A smile is the expression that you have on your face when you are pleased or amused, or when you are being friendly.*

Figure 3.3: Sample text pair from text similarity corpus, classified by 7 out of 10 workers as 1 on a scale of 1-5.

Dataset	IAA	# classes	% rare	Label type	Size
ECD	pctg 0.94	3	34% (artificial)	nominal	100 email pairs
ETP-GOLD	pctg 0.66	3	17% (artificial)	nominal	750 turn/edit pairs
SENTPAIRS	Spearman p=0.91	5	varies	ordinal	30 sentence pairs

Table 3.1: Statistics of the three datasets.

and 3.6) use the most frequent label as the gold standard. This occasionally results in multiple instances derived from one set of ratings, such as the labelset [1,1,1,1,2,4,4,4,4,5] where *1* and *4* occur with equal top frequency.

Although this task could be treated as a multi-class classification or regression problem, we choose to treat it as a series of class-imbalanced binary classification tasks. From the view of binary classification, each one of the 5 classes constitutes a rare class. For the purposes of our experiments, we treat each class in turn as the rare class, while neighboring classes are treated as *can't tell* classes (with estimated normalization for continuum edge classes *1* and *5*), and the rest as common-class instances. For example, experiments treating class *4* as rare treated classes *3* and *5* as "*can't tell*" and classes *1* and *2* as common.

Table 3.1 provides a summary of the three datasets.

3.4 Baseline Costs

Some natural language tasks use corpora that are very class-imbalanced. In a task clustering dictionary definitions, Parent and Eskenazi (2010) obtained crowdsource labels for pairs of definitions; most definition pairs were negative. The evaluation was conducted on a sample of 5 words, but it was noted that the method was unscalable for real-life dictionaries. Pairwise adjacency recognition, which we describe in Chapter 7 and Chapter 8, classifies all pairs of turns within a discussion thread, most of which are non-adjacent.

The ECD and ETP-GOLD datasets consist of Cartesian-product text pairs (within the corpus and within a discussion, respectively), in which nearly all pairs are *negative*. Although the dataset for text similarity rating does not require such pairing, it is still heavily class-imbalanced.

The class imbalance for the The Ecd is as follows. Consider an email corpus with a set of threads T and each $t \in T$ consisting of a set of emails E_t, where rare-class instances are pairs of emails from the same thread, and common-class instances are pairs of emails from different threads. We have the following number of rare-class instances:

$$|\text{ Instances }_{\text{rare}}| = \sum_{i=1}^{|T|} \sum_{j=1}^{|E_i|-1} j$$

and number of common-class instances:

$$|\text{ Instances }_{\text{common}}| = \sum_{i=1}^{|T|} \sum_{j=1}^{|E_i|} \sum_{k=(i+1)}^{|T|} |E_k|$$

For example, in an email corpus with 2 threads of 2 emails each, 4 (67%) of pairs are common-class instances, and 2 (33%) are rare-class instances. If another email thread of two emails is added, 12 (80%) of the pairs are common-class instances, and 3 (20%) are rare-class instances.

These formula show that the contents of a typical corporate email server will have rare-class frequency of well below 0.0001, but that also varies significantly based on small variation of corpus size. To provide a constant value for the purposes of this work, we standardize rare class frequency to 0.01 unless otherwise noted. This is different from our datasets' actual class imbalances, but the conclusions from our experiments in Section 3.6 are independent of class balance.

3.4.1 Baseline Cost

The baseline aggregation technique in our experiments (see Sections 3.5 and 3.6) is majority vote of the annotators. For example, if an instance receives at least 3 out of 5 rare-class labels, then the baseline consensus declares it rare-class. In each of our three corpora, an annotated rare-class instance is expensive because so many common-class instances must be annotated for each rare-class instance as a *by-product*.

Ecd Cost For our Ecd dataset, we solicited 10 *Amazon Mechanical Turk* (MTurk)[38] labels for each of 100 pairs of emails, at a cost of US$0.033[39] per label. Standard quality measures employed to reduce spam labels included over 2000 *HITs* (MTurk tasks) completed, 95% HIT acceptance rate, and location in the US.

[38] `www.mturk.com`

[39] Including approx. 10% MTurk fees

Assuming 0.01 rare-class frequency[40] and 5 labels[41], the cost of a rare-class instance is:

$$\frac{\text{US\$0.033} \times \text{5 annotators}}{\text{0.01 freq}} = \text{US\$16.50}$$

ETP-GOLD Dataset Cost For our ETP-GOLD dataset, we solicited five MTurk labels for each of 750 turn/edit text pairs at a cost of US$0.044 per label. Measures for Wikipedia turn/edit pairs included 2000 HITs completed, 97% acceptance rate, age over 18, and either pre-approval based on good work in pilot studies or a high score on a qualification test of sample pairs. The cost of a rare-class instance is:

$$\frac{\text{US\$0.044} \times \text{5 annotators}}{\text{0.01 freq}} = \text{US\$22}$$

SENTPAIRS Dataset Cost The SENTPAIRS dataset consists of 30 sentence pairs, and 10 labels per pair. The original price of Bär et al. (2011)'s sentence pairs corpus is unknown, so we estimated a cost of US$0.01 per label. The labels came from Crowdflower[42]. Bär et al. (2011) used a number of quality assurance mechanisms, such as worker reliability and label correlation. The cost of a rare-class instance varied between classes, due to class frequency variation, from instance_{class2}=US$0.027 to instance_{class5}=US$0.227.

Finding versus Confirming a Rare-class Instance It is cheaper to confirm a rare-class instance than to find a suspected rare-class instance in the first place. The task presents two types of binary decisions: finding a suspected rare-class instance ("Is the instance a true positive (*TP*) or false negative (*FN*)?") and confirming a rare-class instance as rare ("Is the instance a TP or false positive (*FP*)?"). Assuming a 0.01 rare-class frequency, 5-label majority-vote decision, and 0.5 FP frequency, the cost of the former is:

$$\frac{\text{1 label}}{\text{0.01 freq}} + \frac{\text{1 label}}{\text{0.99 freq}} =\sim \text{101 labels}$$

and the latter is:

$$\frac{\text{5 labels}}{\text{0.5 freq}} = \text{10 labels}$$

In this scenario, the per-instance cost of finding a rare-class instance is ~101 labels, while the per-instance cost of confirming a rare-class instance is only 10 labels. Thus, we focus

[40] Although this chapter proposes a hypothetical 0.01 rare-class frequency, the ECD and ETP-GOLD datasets have been partially balanced: the negative instances merely functioned as a distractor for annotators, and conclusions drawn from the rule cascade experiments only apply to positive instances.

[41] On this dataset, IAA was high and 10 labels was over-redundant.

[42] `crowdflower.com`

our work on reducing the costs of *finding* rare-class instances, because it is so much more expensive than *confirming* rare-class instances.

Metrics We used the following metrics for our experiment results:
TP is the number of true positives (rare-class) discovered. The fewer TP's discovered, the less likely the resulting corpus will represent the original data in an undistorted manner.
$\mathbf{P}_{rare}$ is the precision over rare instances: $\frac{TP}{TP+FP}$ Lower precision means lower confidence in the produced dataset, because the "rare" instances we found might have been misclassified.
AvgA is the average number of labels needed for the system to make a common-class judgment, i.e. to stop investigation of the instance as rare-class. AvgA allows comparison between annotation tasks without task-specific HIT cost and rare-class prevalence. The lower bound for AvgA is 1.0 (i.e., each instance requires at least one label).
The normalized cost is the estimated cost of acquiring a rare instance: $\frac{\frac{\text{AvgA}\times\text{annoCost}}{\text{classImbalance}}}{\text{Recall}_{rare}}$
Savings is the estimated financial cost saved when identifying rare instances, over the baseline. Includes Standard Deviation.

3.5 Supervised Cascading Classifier Experiments

Previous work (Zaidan and Callison-Burch, 2011) used machine learners trained on crowdsourcing label metadata (e.g., worker's country of residence and work time duration on the HIT, as associated with a worker assigning a label to an instance) to predict crowdsource label quality, so that only bad-quality annotations needed to be repetitively labeled. In this section, we used crowdsourcing label metadata as features for a cascading logistic regression classifier to choose whether or not an additional redundant label is needed. In each of the five cascade rounds, an instance was machine classified as either *potentially rare* or *common*. Instances classified as potentially rare received a repeat label and continued through the next cascade, while instances classified as common were discarded. Discarding instances before the end of the cascade can reduce the total number of needed labels (AvgA), and therefore lower annotation cost. This cascade models the observation (see "Finding versus Confirming a Rare-class Instance", Section 3.4.1) that it is cheap to confirm suspected rare-class instances, but it is expensive to weed out common-class instances, by quickly discarding instances that the classifier is confident are not rare-class, but spending additional resources as necessary to confirm rare-class instances.

Experiments from this section will be compared in Section 3.6 to a rule-based cascading classifier system that, unlike this supervised system, does not need any training data.

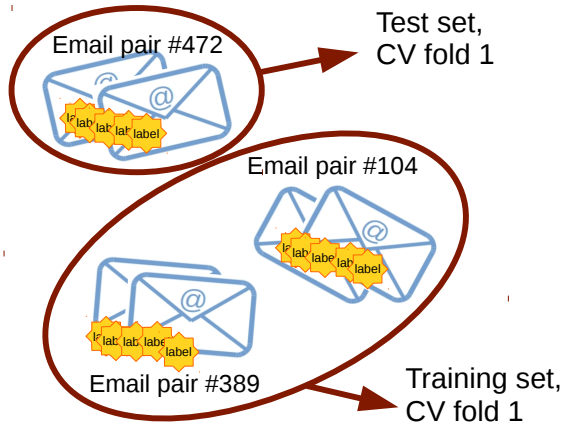

Figure 3.4: Cross-validation fold division: text pairs were assigned to the training or test set within a fold.

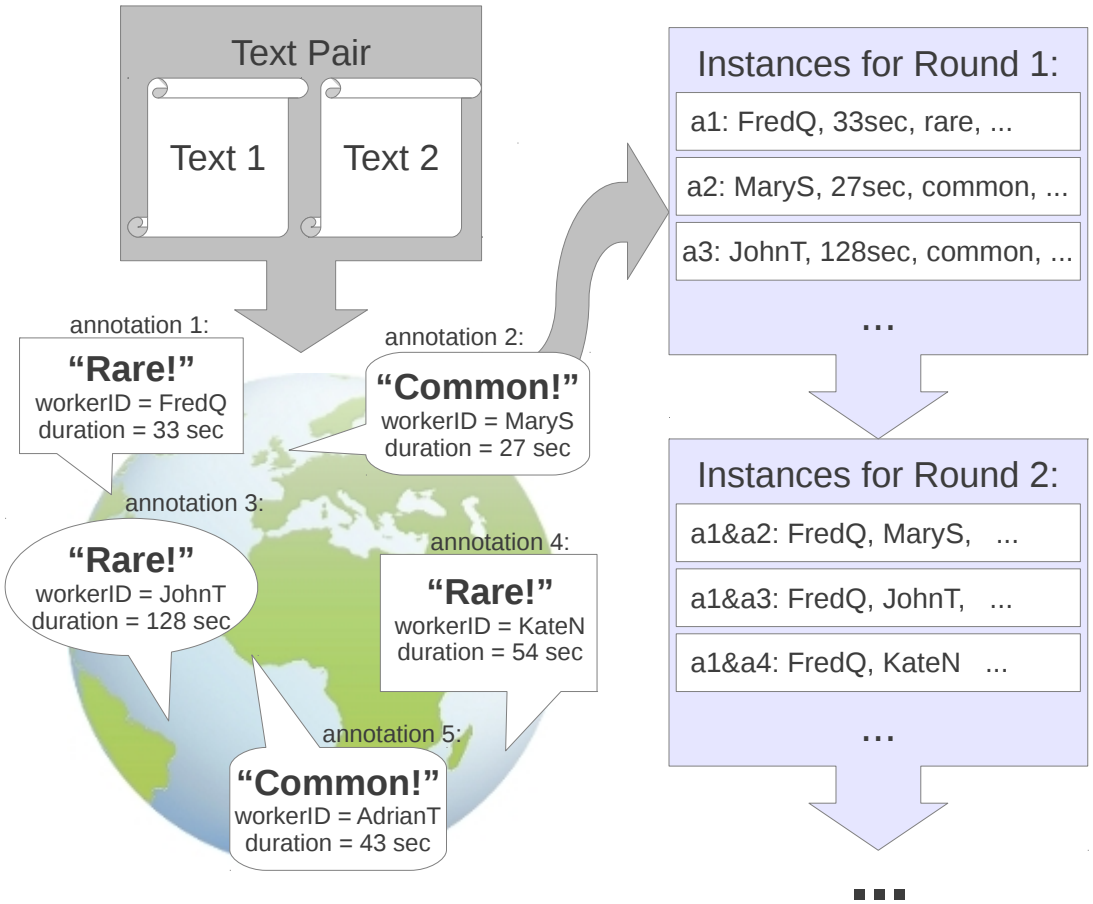

Figure 3.5: Multiple learning instances are generated from each original annotated text pair.

3.5.1 Instances

Each experimental instance consisted of features derived from the *metadata* of one or more crowdsourced labels from a pair of texts. A gold standard rare-class instance has >80% rare labels, i.e., an instance labeled by five workers was unanimous.

In the first round of experiments, each instance was derived from a single label. In each further round, instances were only included that consisted of an instance from the previous round that had been classified *potentially rare* plus one additional label. All possible instances were used that could be derived from the available annotations, as long as the instance was permitted by the previous round of classification, resulting in potentially multiple instances per original text pair for all rounds except the final round. Figure 3.5 shows how multiple instances were generated from a single text pair. 10-fold cross-validation was used, with fold division of text pairs to avoid evaluation on pair-specific features, as shown in Figure 3.4. For further discussion of related information leak issues in a cross-validation paradigm, please see Chapter 7.8.

Although SentPairs had 10 labels per pair, we stopped the supervised cascade at five iterations, because the number of rare-class instances was too small to continue. This resulted in a larger number of final instances than actual sentence pairs.

3.5.2 Features

Features were derived from the *metadata* of the label and its HIT. Features included a label's worker ID, estimated time duration, completion day of the week (Ecd and Etp-Gold only), and the label (*rare*, *common*, *can't tell*), as well as all possible joins of one label's features (`yesAND-JohnTAND30sec`). For instances representing more than a single label, a feature's *count* over all the labels was also included (i.e., `common:3` for an instance including 3 *common* labels). For reasons discussed in Section 3.1, we exclude features based on text content of the pair. Figure 3.2 provides an overview of these features.

3.5.3 Results

Tables 3.3 and 3.4 show the results of our trained cascading system on Ecd and Etp-Gold, respectively; baseline is majority voting. Tables 3.5 and 3.6 show results on rare classes `1` and `5` of SentPairs (classes `2`, `3`, and `4` had too few instances to train, a disadvantage of a supervised system that is fixed by our rule-based system in Section 3.6). The baseline was previously described in Section 3.4.

Table 3.3 shows that the best feature combination for identifying rare email pairs was label, worker ID, and day of the week ($4.35 per rare instance, and 33/34 instances found); however, this was only marginally better than using label alone ($4.68, 31/34 instances found). The best feature combination resulted in a 74% cost savings over the baseline.

Feature	Definition	Example	Motivation
worker	the ID of the annotator	"A37WXDYYT748"	Certain workers may be more reliable than others.
dur	endTime - startTime of the HIT task	49 seconds	Very fast workers might be clicking buttons without doing the work.
day	day of the week of HIT task completion	"Tuesday"	Workers might do better work on a Tuesday afternoon while at work than on a Saturday morning recovering from last night's party.
label	class assigned by the annotator	"yes", "no"	An instance with a "yes" label is more likely to be a true gold "yes" (positive) instance, although this depends on the reliability of this annotation
joins	all possible combinations of features from a single annotation	yesANDJohnTANDTuesday	This is to model the reliability of a single annotation, without creating a sparse unusable feature space of all possible feature combinations.

Table 3.2: Raw metadata features used for supervised cascade.

features	TP's	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	34	1.00	-	$16.50	-
label	31	0.88	1.2341	$4.68	72±8
worker	0	0.0	1.0	-	-
dur	2	0.1	1.0	$16.5	0±0
day	0	0.0	1.0	-	-
worker & label	33	0.9	1.1953	$4.38	73±7
day & label	31	0.88	1.2347	$4.68	72±8
dur & label	33	0.88	1.2437	$4.56	72±8
w/o label	3	0.12	1.2577	$20.75	-26±41
w/o worker	33	0.9	1.2341	$4.53	73±8
w/o day	33	0.9	1.2098	$4.44	73±7
w/o dur	33	0.9	1.187	**$4.35**	**74±7**
all	33	0.9	1.2205	$4.48	73±8

Table 3.3: ECD results on the supervised cascade.

features	TP's	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	128	1.00	-	$22.00	-
label	35	0.93	1.7982	$20.29	08±32
worker	0	0.0	1.0	-	-
dur	0	0.0	1.0	-	-
day	0	0.0	1.0	-	-
worker & label	126	0.99	1.6022	**$7.12**	**68±11**
day & label	108	0.88	1.644	$8.51	61±13
dur & label	111	0.86	1.5978	$8.08	63±12
w/o label	4	0.12	1.0259	$11.28	49±6
w/o worker	92	0.84	1.7193	$9.46	57±15
w/o day	104	0.9	1.6639	$8.61	61±14
w/o dur	109	0.94	1.6578	$8.2	63±14
all	89	0.82	1.6717	$8.76	60±15

Table 3.4: ETP-GOLD results on the supervised cascade.

features	TP's	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	12	1.00	-	$1.50	-
label	9	0.67	1.8663	$0.4	73±10
workerID	1	0.1	1.5426	$2.31	-54±59
dur	2	0.15	1.4759	$1.11	26±26
worker & label	11	0.7	1.8216	**$0.39**	**74±9**
worker & dur	3	0.2	1.8813	$1.41	06±34
dur & label	8	0.42	1.8783	$0.56	62±13
all	11	0.62	1.8947	$0.41	73±8

Table 3.5: SENTPAIRS$_{c1}$ results on the supervised cascade.

features	TP's	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	17	1.00	-	$0.44	-
label	14	0.72	2.4545	$0.15	66±7
worker	14	0.63	2.7937	$0.16	64±8
dur	10	0.52	2.7111	$0.18	58±11
worker & label	15	0.82	2.3478	**$0.12**	**73±8**
worker & dur	6	0.4	2.7576	$0.38	14±23
dur & label	16	0.72	2.4887	$0.14	69±10
all	17	0.82	2.4408	$0.12	73±5

Table 3.6: SENTPAIRS$_{c5}$ results on the supervised cascade.

Rule	Explanation
no >0	"If there are any *no* (i.e., common) labels, the instance is assumed to be common and no further labels are obtained."
no >1	"If there are more than 1 *no* (i.e., common) labels, the instance is assumed to be common and no further labels are obtained."
no >2	"If there are more than 2 *no* (i.e., common) labels, the instance is assumed to be common and no further labels are obtained."
(no+ct) >0	"If there are any *no* (i.e., common) labels or any *can't tell* labels, the instance is assumed to be common and no further labels are obtained."
(no+ct) >1	"If there are more than 1 total *no* (i.e., common) or *can't tell* labels, the instance is assumed to be common and no further labels are obtained."
(no+ct) >2	"If there are more than 2 total *no* (i.e., common) or *can't tell* labels, the instance is assumed to be common and no further labels are obtained."

Table 3.7: Explanation of rules used in the rule-based cascade.

Table 3.4 shows that the best feature combination for identifying rare Etp-Gold pairs was label and worker ID ($7.12, 126/128 instances found). Unlike the Ecd experiments, this combination was remarkably more effective than labels alone ($20.29, 35/128 instances found), and produced a 68% total cost savings.

Tables 3.5 and 3.6 show that the best feature combination for identifying rare sentence pairs for both rare classes 1 and 5 was also label and worker ID (US$0.39 and US$0.12, respectively), which produced a 73% cost savings; for class 5, adding the duration feature minimally decreased the standard deviation of the savings. Label and worker ID were only marginally better than label alone for class 1.

As can be seen in Tables 3.3, 3.4, 3.5, and 3.6, nearly all the True Positives were found with the respective best performing systems: only a total of 4 out of 191 TP's were missed by the cascades, and the average number of needed labels for the best Tables 3.3, 3.4, and 3.5 systems is fewer than 2. The critical room for improvement lies in being able to find rare-class instances using an unsupervised system, so that no expensive training data is required. Therefore, in the next section, we investigate a rule-based cascade approach.

3.6 Rule-based Cascade Experiments

Although the metadata-trained cascading classifier system is effective in reducing the needed number of labels, it is not useful in the initial stage of annotation, when there is no training data. In these experiments, we evaluate a *rule-based cascade* in place of our previous supervised classifier. The rule-based cascade functions similarly to the trained classifier cascade except that a single rule replaces each classification. Five cascades are used.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	128	1.00	-	$22.0	-
no > 0	39	0.95	1.61	**$7.09**	**68±16**
no > 1	39	0.85	2.86	$12.6	43±19
no > 2	39	0.73	3.81	$16.75	24±15
(no+ct) > 0	22	0.98	1.35	$10.56	52±20
(no+ct) > 1	33	0.93	2.55	$13.25	40±18
(no+ct) > 2	35	0.85	3.56	$17.44	21±15

Table 3.8: ETP-GOLD results: rule-based cascade. All instances included.

Each rule instructs when to discard an instance, avoiding further labeling cost. For example, `no>2` means, "if the count of *no* (i.e., common) labels becomes greater than 2, we assume the instance is common and do not seek further confirmation from more labels." The gold standard is the same as the supervised cascade in Section 3.5. As the rule-based cascade is unsupervised, and there were no parameters to tune, all data is used in the test set.

For our rule-based experiments, we define AvgA for each instance i and for labels a_{1_i}, a_{2_i}, ..., a_{5_i} and the probability (Pr) of five rare-class labels. Class c is the common class. We always need a first label: $\Pr(a_{1_i} \neq c) = 1$.

$$AvgA_i = \sum_{j=1}^{5} \prod_{k=1}^{j} \Pr(a_{k_i} \neq c)$$

We define $Precision_{rare}$ (P_{rare}) as the probability that instance i with 5 common[43] labels a_{1_i}, a_{2_i}, ..., a_{5_i} is not a rare-class instance:

$$\begin{aligned} P_{rare_i} &= \Pr(\text{TP}|(a_{1...5_i} = \text{rare})) \\ &= 1 - \Pr(\text{FP}|(a_{1...5_i} = \text{rare})) \end{aligned}$$

Thus, we estimate the probability of seeing other FP's based on the class distribution of our labels. This is different from our supervised cascade experiments, in which $P_{rare} = \frac{TP}{TP+FP}$.

3.6.1 Results

Table 3.8 shows the results of various rule systems on reducing cost on the ETP-GOLD data.

While it might appear reasonable to permit one or two careless crowdsource labels before discarding an instance, the tables show just how costly this allowance is: each permitted extra label (i.e., `no>1`, `no>2`, ...) must be applied systematically to each instance (because we do not

[43]This may also include *can't tell* labels, depending on the experiment.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	128	1.00	-	$22.0	-
no > 0	35	0.96	1.46	**$6.43**	**71±14**
no > 1	35	0.9	2.67	$11.76	47±17
no > 2	35	0.81	3.66	$16.11	27±14
(no+ct) > 0	22	0.98	1.33	$9.34	58±19
(no+ct) > 1	33	0.92	2.5	$11.66	47±17
(no+ct) > 2	35	0.85	3.49	$15.36	30±13

Table 3.9: ETP-GOLD results: no ambiguous instances.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	34	1.00	-	$16.5	-
no > 0	32	1.0	1.07	**$3.52**	**79±6**
no > 1	32	0.99	2.11	$6.95	58±7
no > 2	32	0.98	3.12	$10.31	38±6
(no+ct) > 0	30	1.0	1.04	$3.67	78±5
(no+ct) > 1	32	0.99	2.07	$6.83	59±6
(no+ct) > 2	32	0.99	3.08	$10.16	38±5

Table 3.10: ECD results: rule-based cascade.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	5	1.00	-	$1.5	-
no > 0	5	0.99	1.69	**$0.25**	**83±10**
no > 1	5	0.96	3.27	$0.49	67±17
no > 2	5	0.9	4.66	$0.7	53±21
(no+ct) > 0	0	1.0	1.34	-	-
(no+ct) > 1	2	0.98	2.63	$0.98	34±31
(no+ct) > 2	4	0.96	3.83	$0.72	52±19

Table 3.11: SentPairs_{c1} results: rule-based cascade.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	2	1.00	-	$3.75	-
no > 0	2	0.98	1.95	**$0.73**	**81±12**
no > 1	2	0.93	3.68	$1.38	63±20
no > 2	2	0.86	5.12	$1.92	49±23
(no+ct) > 0	0	1.0	1.1	-	-
(no+ct) > 1	0	1.0	2.2	-	-
(no+ct) > 2	0	1.0	3.29	-	-

Table 3.12: SENTPAIRS_{c2} results: rule-based cascade.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	16	1.00	-	$0.47	-
no > 0	16	0.99	1.98	**$0.09**	**80±9**
no > 1	16	0.96	3.83	$0.18	62±15
no > 2	16	0.9	5.47	$0.26	45±17
(no+ct) > 0	0	1.0	1.23	-	-
(no+ct) > 1	0	1.0	2.45	-	-
(no+ct) > 2	1	0.99	3.65	$2.74	-484±162

Table 3.13: SENTPAIRS$_{c4}$ results: rule-based cascade.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	17	1.00	-	$0.44	-
no > 0	17	0.99	1.96	**$0.09**	**80±10**
no > 1	17	0.95	3.77	$0.17	62±16
no > 2	17	0.89	5.37	$0.24	46±18
(no+ct) > 0	2	1.0	1.27	$0.48	-8±21
(no+ct) > 1	10	1.0	2.54	$0.19	57±8
(no+ct) > 2	13	1.0	3.8	$0.22	50±9

Table 3.14: SENTPAIRS$_{c5}$ results: rule-based cascade.

know which labels are careless and which are accurate) and can increase the average number of labels needed to discard a common instance by over 1. The practice also decreases rare-class precision, within an *n*-labels limit. Clearly the cheapest and most precise option is to discard an instance as soon as there is a common-class label.

When inherently ambiguous instances are shifted from rare to common by including *can't tell* as a common-class label, the cost of a rare ETP-GOLD instance falls from US$7.09 (68% savings over baseline) to US$6.10 (72% savings), and the best performing rule is `(no+ct)>0`. A rare email instance barely increases from US$3.52 (79% savings) to US$3.65 (78% savings). However, in both cases, TP of rare-class instances falls (ETP-GOLD: 39 instances to 22, ECD: 32 instances to 30). This does not affect overall cost, because it is already included in the equation, but the rare-class instances found may not be representative of the data.

There was not much change in precision in the ETP-GOLD dataset when *can't tell* was included as a rare-class label (such as `no>0`) or a common-class label (such as `(no+ct)>0`), so we assume that the populations of rare instances gathered are not different between the two. However, when a reduced number of TP's are produced from treating *can't tell* as a common label, higher annotation costs can result (such as Table 3.8, `no>0` cost of US$7.09, versus `(no+ct)>0` cost of US$10.56).

Removing ambiguous instances from the test corpus does not notably change the results (see Table 3.9). We defined ambiguous instances as those where the majority class was *can't*

tell, the majority class was tied with *can't tell*, or there was a tie between common and rare classes.

Finally, the tables show that not only do the top-performing rules save money over the 5-labels baseline, they save about as much money as supervised cascade classification.

Table 3.10 shows results from the Ecd dataset. Results largely mirrored those of the Etp-Gold dataset, except that there was higher inter-annotator agreement on the email pairs which reduced annotation costs. We also found that, similarly to the Etp-Gold experiments, weeding out uncertain examples did not notably change the results.

Results of the rule-based cascade on SentPairs are shown in Tables 3.11, 3.12, 3.13, and 3.14. There were no instances with a most frequent class 3. Also, there are more total rare-class instances than sentence pairs, because of the method used to identify a gold instance: labels neighboring the rare class were ignored, and an instance was gold rare if the percentage of rare labels was >0.8 of total labels. Thus, an instance with the count {class1=5, class2=4, class3=1, class4=0, class5=0} counts as a gold instance of both class 1 and class 2.

The cheapest rule was `no>0`, which had a recall of 1.0, P_{rare} of 0.9895, and a cost savings of 80-83% (across classes 1-5) over the 10 annotators originally used in this task.

3.6.2 Error Analysis

A rare-class instance with many common labels has a greater chance of being classified common-class and thus discarded by a single crowdsource worker screening the data. What are the traits of rare-class instances at high risk of being discarded? We analyzed only Etp-Gold text pairs, because the inter-annotator agreement was low enough to cause false negatives. The small size of SentPairs and the high inter-annotator agreement of Ecd prevented analysis.

Etp-Gold data The numbers of instances (750 total) with various crowdsource label distributions are shown in Table 3.15. The table shows label distributions (i.e., `302` = 3 *yes*, 0 *can't tell* and 2 *no*) for high and low agreement rare-class instance counts. An instance with low agreement has a higher probability of being missed, due to chance variation of which worker labels the instance first: an instance with 2 total *no* labels has a 40% chance of being discarded in the first round of a *no >0* rule cascade, while and instance with 1 total *no* label only has a 20% chance.

We analyzed the instances from the category most likely to be missed (`302`) and compared it with the two categories least likely to be missed (`500`, `410`). Of five random `302` pairs, all five appeared linguistically ambiguous and difficult to annotate; they were missing discussion context that was known (or assumed to be known) by the original participants. Two of the turns state future deletion operations, and the edits include deleted statements, but it is unknown if the turns were referring to these particular deleted statements or to others. In another instance, the turn argues that a contentious research question has been answered and that the

Ambiguous instances		Unambiguous instances	
Labels	Number	Labels	Number
y ct n	instances	*y ct n*	instances
3 0 2	35	5 0 0	22
3 1 1	30	4 1 0	11
2 2 1	19	4 0 1	28
2 1 2	39	3 2 0	2

Table 3.15: Label distributions and instance counts from ETP-GOLD.

user will edit the article accordingly, but it is unclear in which direction the user intended to edit the article. In another instance, the turn requests the expansion of an article section, and the edit is an added reference to that section. In the last pair, the turn gives a quote from the article and requests a source, and the edit adds a source to the quoted part of the article, but the source clearly refers to just one part of the quote.

In contrast, we found four of the five `500` and `410` pairs to be clear rare-class instances. Turns quoted text from the article that matched actions in the edits. In the fifth pair, a `500` false positive instance, the edit was first made, then the turn was submitted complaining about the edit and asking it to be reversed. This was a failure by the annotators to follow the directions included with the task describing permitted types of positive labels.

3.7 Chapter Summary

In this chapter, we presented two approaches to reduce annotation costs on class-imbalanced corpora. In the first approach, we used a supervised machine learner cascade to classify an instance as being rare class based solely on the instance's metadata, so that downstream usage would not be limited by selection bias of the rare class. In the second approach, we created a rule-based cascade to classify an instance as being rare-class, also based solely on the instance's metadata, but with the additional benefit that, as an unsupervised technique, no training corpus is required.

We answered the following questions, posed at the beginning of this chapter:

> *Research Question*: It has been shown that annotation quality on a class-balanced dataset is improved by redundant labeling. Should a class-imbalanced dataset be redundantly crowd-labeled?

Our findings show that, no, a class-imbalanced dataset should **not** be redundantly labeled, because redundant labels are very costly in a class-imbalanced annotation task. Specifically, we found that an instance that has received a single common-class label should be presumed to be common-class, and should be discarded during the search for rare-class instances. We showed this to be the case for all six rare classes in our three class-imbalanced datasets.

> *Research Question*: How cost effective is discarding instances that receive a single common-class label, compared to a trained, metadata-feature-based classifier cascade, to acquire rare-class instances?

Although our rule-based technique of discarding instances that receive a single common-class label, during the search for rare-class instances, is radically simpler than our previously proposed technique of identifying rare-class instances using a supervised machine classifier cascade trained on instance metadata, we have found both techniques have roughly the same cost, which is about 70% cheaper than the baseline 5-vote majority vote aggregation. Further, the rule cascade requires no training data, making it suitable for seed dataset production.

Class-imbalanced datasets are found in a variety of other natural language processing tasks; in particular, any task that creates graphs or clusters of texts, and which requires pairwise text annotations by humans, will use a class-imbalanced corpus. Any experiment for such a task that uses a supervised machine classifier will require a certain minimum number of human-annotated rare-class instances to learn a model, and any unsupervised machine classifier will require this for evaluation; such experiments will face the expensive annotation scenario described in this chapter. Future work will be necessary to investigate the applicability of our findings to other class-imbalanced datasets.

In this chapter, we have contributed solutions towards problems faced in annotating a discussion thread corpus for our target task of thread reconstruction. In Chapter 4, we investigate techniques to best utilize our crowdsource labels to train a machine classifier, using a variety of other, well-established natural language tasks. The current chapter, combined with Chapter 4, enable thread reconstruction research, as well as research on other natural language tasks with class-imbalanced or crowdsource-labeled corpora, by solving associated annotation problems encountered while creating a corpus for this task.

Chapter 4

Text Classification of Crowdsourced Datasets

Thread reconstruction is a relatively new natural language task, and there are not many pre-existing datasets suitable for reconstruction experiments. The advent of crowdsourcing (described in Chapter 2) has greatly reduced the cost and resource requirements of corpus annotation, making possible the creation of corpora for new NLP tasks such as thread reconstruction. Previously, in Chapter 3, we explained annotation cost problems associated with class-imbalanced corpora for tasks such as thread reconstruction, and we provided a solution to reduce such costs. However, crowdsource labels are known to be noisy (more than expert human annotators (Snow et al., 2008)), and it is not clear how to derive a gold standard from these noisy labels that can be used to train a machine classifier. Although a thread reconstruction corpus that was crowdsource annotated using the techniques described in Chapter 3 would have unanimous rare-class annotations, the common-class annotations may contain noise. In order to investigate thread reconstruction via the text classification experiments in Chapters 6, 7, and 8, we must train a classifier on the labels.

In this chapter, we bridge the gap from a newly created crowdsource-annotated corpus, to automatic text classification, by investigating how to learn the best machine classifier model from a set of crowdsourced labels. To provide a generalized view beyond thread reconstruction alone, and to ensure that unexpected system performance in brand new NLP tasks does not affect the results, we investigate five different and well-established natural language tasks. For each task, we examine the impact of passing item agreement on to the task classifier, by means of soft labeling, and of changing the training dataset via low-agreement filtering. We address the following research questions:

> *Research Question*: In the context of crowdsourced datasets, is the best classifier produced from a training dataset of integrated labels, or from item agreement filtering, or from soft labeling?

Research Question: Some instances are naturally more difficult for humans to decide on annotations for. Do our strategies, as listed above, have an equal impact on both Hard Cases and Easy Cases in the test data?

Research Question: How does corpus size impact performance of training strategy: Which training strategy performs best with different size corpora? What is the added benefit of additional high-agreement training instances compared to additional generic training instances?

The chapter is structured as follows. First, we provide an overview of our motivation (Section 4.1), and a discussion of previous research (Section 4.2). We provide an overview of our experiments (Section 4.3), and then we present five natural language processing tasks: biased language detection (Section 4.4), stemming classification (Section 4.5), recognizing textual entailment (Section 4.6), Twitter POS tagging (Section 4.7), and affect recognition (Section 4.8). In the section for each task, we discuss the effects on their classifiers when trained with item agreement filtering and soft labeling, compared with *integrated labels* (majority vote or mean baseline, respective to the task). We conclude the chapter with a summary of our findings (Section 4.9).

Most of the material in this chapter was previously published in peer-reviewed proceedings:

Emily K. Jamison and Iryna Gurevych: 'Noise or additional information? Using crowdsource annotation item agreement for natural language tasks', in: *Proceedings of EMNLP 2015: Conference on Empirical Methods in Natural Language Processing* (EMNLP 2015), p. 291–297, Lisbon, Spain, 2015. (Chapter 4)

4.1 Motivation

Crowdsourcing is a cheap and increasingly-utilized source of annotation labels. In a typical annotation task, five or ten labels are collected for an instance, and are aggregated together, using techniques such as majority voting or latent-variable modeling, into an *integrated label*. The high number of labels is used to compensate for worker bias, task misunderstanding, lack of interest, incompetence, and malicious intent (Wauthier and Jordan, 2011).

The goal of label aggregation for producing gold labels is to reduce noise in the training data, with the expectation that this will produce a more accurate machine classifier. Majority Voting has been found effective in filtering noisy labels (Nowak and Rüger, 2010). Labels can be aggregated under weighted conditions reflecting the reliability of the annotator (Whitehill et al., 2009; Welinder et al., 2010). Certain classifiers are also robust to random (unbiased) label noise (Tibshirani and Manning, 2014; Beigman and Klebanov, 2009). However, minority label

Text: *The prosecutor told the court that the incident had caused "distress" to one of the children.*
Hypothesis: *The prosecutor told the court that "distress" in one of the children is ascribed to the incident.*
Entailment Status: True
α **Agreement (-1.0 – 1.0)**: 1.0

Figure 4.1: RTE Easy Case.

Text: *Bush returned to the White House late Saturday while his running mate was off campaigning in the West.*
Hypothesis: *Bush left the White House.*
Entailment Status: True
α **Agreement (-1.0 – 1.0)**: -0.1

Figure 4.2: RTE Hard Case.

information is discarded by majority voting, and when the labels were gathered from screened and qualified crowdsource workers, the noise in the labels may contain useful information.

Consider the two textual entailment instances and their Krippendorff (1970)'s *α item agreement*, shown in Figures 4.1 and 4.2. In Figure 4.1, the annotators all agreed that the label should be *true*. In Figure 4.2, the low item agreement (does *Bush returned* entail *Bush left*?) might be caused by worker bias, unreliability, or malicious intent, but the instance may also simply be difficult: a *Hard Case*. When Hard Case labels are aggregated, minority-label information is lost. Linguistic ambiguity of test cases may also be impacted: in a semantic newness classification task, Beigman Klebanov and Beigman (2014) showed that training strategy may affect Hard and Easy Case test instances differently, with the presence of Hard Cases in the training data leading to the misclassification of Easy Cases in the test data.

Two alternative strategies that allow the classifier to learn from the item agreement include training instance *filtering* and *soft labeling*. Filtering training instances by item agreement removes low agreement instances from the training set. Soft labeling assigns a classifier weight to a training instance based on the item agreement.

In this chapter, for five natural language tasks, we examine the impact of passing crowdsource item agreement on to the task classifier, by means of training instance filtering and soft labeling. We construct classifiers for Biased Text Detection, Stemming Classification, Recognizing Textual Entailment, Twitter POS Tagging, and Affect Recognition, and evaluate the effect of our different training strategies on the accuracy of each task. The five natural language tasks in this chapter were chosen for the diverse nature of statistical natural language tasks: sentence-level linear regression using n-grams; word pairs with character-based features and binary linear classification; pairwise sentence binary linear classification with similarity score features; CRF sequential word classification with a range of feature types; and sentence-level

regression using a token-weight averaging, respectively. We use pre-existing, freely-available crowdsourced datasets[44] and post all our experiment code on GitHub[45].

Contributions While previous research has examined the impact of agreement filtering or soft labeling on a single task (Beigman Klebanov and Beigman, 2014; Plank et al., 2014; Martínez Alonso et al., 2015), the research in this chapter is the first work (1) to apply item-agreement-weighted soft labeling from crowdsourced labels to multiple real natural language tasks; (2) to filter training instances by item agreement from crowdsourced labels, for multiple natural language tasks; (3) to evaluate classifier performance on high item agreement (Hard Case) instances and low item agreement (Easy Case) instances across multiple natural language tasks.

4.2 Previous Work

Much previous work has examined techniques to aggregate training instance labels into an integrated label. Nowak and Rüger (2010) examined crowdsource and expert labels for an image annotation task and determined that majority vote aggregation was effective at removing noisy annotations and that the resulting integrated labels were comparable in quality to expert labels. Dekel and Shamir (2009) calculated integrated labels for an information retrieval crowdsourced dataset, and identified low-quality workers by deviation from the integrated label. Removal of these workers' labels improved classifier performance on data that was not similarly filtered. While a large amount of work (Dawid and Skene, 1979b; Whitehill et al., 2009; Welinder et al., 2010; Ipeirotis et al., 2010; Dalvi et al., 2013; Tang and Lease, 2011) has explored techniques to model worker ability, bias, and instance difficulty when aggregating labels, we note that these integrated labels are not used to train classifiers for their respective NLP tasks. Fiscus (1997) created integrated labels from the opposite end of the system pipeline: they produce a composite Automatic Speech Recognition (ASR) system that combines the predicted labels from multiple independent ASR systems into a single integrated label with lower error rates than any of the independent systems.

Training instance filtering aims to remove mislabeled instances from the training dataset. Brodley and Friedl (1999) trained classifiers to act as filters and identify mislabeled outlier data before it is used as training data, and found that this technique improves classification accuracy for noise levels up to 30%. Sculley and Cormack (2008) learned a logistic regression classifier to identify and filter noisy labels in a spam email filtering task. They also proposed a label correcting technique that replaces identified noisy labels with "corrected" labels, at the risk of introducing noise into the corpus. The techniques are effective with synthetic noise but not natural noise. Rebbapragada et al. (2009) developed a label noise detection technique to

[44] Samples of the datasets, along with item agreement, are available in Appendix A.

[45] `github.com/EmilyKJamison`

cluster training instances and remove label outliers. A classifier trained on the reduced-noise data yielded a 29% false-positive improvement in a sulfur image identification task. Raykar et al. (2010b) extended previous research on modeling annotator accuracy by jointly learning a classifier/regressor, annotator accuracy, and the integrated label on datasets with multiple noisy labels; their joint learning model outperforms Smyth et al. (1995)'s model of estimating ground truth labels followed by classifier training.

Soft labeling, or the association of one training instance with multiple, weighted, conflicting labels, is a technique to model noisy training data. Thiel (2008) found that soft labeled training data produced more accurate classifiers than hard labeled training data, with both Radial Basis Function networks and Fuzzy-Input Fuzzy-Output Support Vector Machines (SVMs). Hoi and Lyu (2004) implemented a soft label SVM for content-based image retrieval, and found that it can better incorporate noise from the users' log-based relevance feedback. Shen and Lapata (2007) used soft labeling to model their semantic frame structures in a question answering task, to represent that the semantic frames can have multiple semantic roles.

Previous research has found that, for a few individual NLP tasks, training while incorporating label noise weight may produce a better model. Martínez Alonso et al. (2015) show that informing a parser of annotator disagreement via loss function reduced error in labeled attachments by 6.4%. Plank et al. (2014) incorporate annotator disagreement in POS tags into the loss function of a POS-tag machine learner, resulting in improved performance on downstream chunking. Beigman Klebanov and Beigman (2014) observed that, on a task classifying text as semantically *old* or *new*, the inclusion of Hard Cases in training data resulted in reduced classifier performance on Easy Cases.

4.3 Experiments Overview

In this chapter, we examine the impact of passing crowdsource item agreement on to the task classifier, by means of training instance filtering, training instance and soft labeling, for five NLP tasks: Biased Text Detection, Stemming Classification, Recognizing Textual Entailment, Twitter POS Tagging, and Affect Recognition. All of our systems except Affect Recognition (which used a unique statistical regression classifier and required minimal feature extraction) were constructed using DKPro TC (Daxenberger et al., 2014), an open-source UIMA-based text classification framework.

Several tasks in this chapter use Support Vector Machines (SVMs). Although Hoi and Lyu (2004) explored a technique to integrate soft labels into an SVM, we follow Sheng et al. (2008)'s *multiplied examples* procedure: for each unlabeled instance x_i and each existing label $l \in L_i = \{y_{ij}\}$, we create one replica of x_i, assign it l, and weight the instance according to the count of l in L_i. All SVM experiments used Weka's *sequential minimal optimization* (SMO) (Platt, 1998) or SMOreg for regression (Shevade et al., 1999) implementations with default parameters. Nominal classification results are reported in micro-F_1, and numerical regression

results are reported in Pearson correlation r. Nominal classification statistical significance is reported using paired TTest, and numerical regression statistical significance is reported using McNemar's Test[46] (McNemar, 1947).

In the training datasets of our experiments, Krippendorff (1970)'s α item agreement was used to filter ambiguous training instances. Filter cutoffs were optimized via micro F_1 using a development set. For soft labeling, percentage item agreement was used to assign instance weights. Please note that evaluation datasets were the same regardless of training strategy, and were carefully integrated with the cross-validation; only the training dataset was changed.

Integrated baseline When labels were numeric, the integrated label was the average[47]. When labels were nominal, the integrated label was majority vote.

Percentage agreement We evaluate on test sets divided by test instance item agreement. In previous work, Whitehill et al. (2009) jointly model instance difficulty along with true label and annotator expertise in their latent-variable algorithm. Beigman Klebanov and Beigman (2014) obtain 20 crowdsource labels for each instance, and assign item agreement categories based on percentage agreement of the labels. We follow Beigman Klebanov and Beigman (2014) in using the nominal agreement categories Hard Cases and Easy Cases to separate instances by item agreement. However, unlike Beigman Klebanov and Beigman (2014) who use simple percentage agreement, we calculate item-specific agreement via Krippendorff (1970)'s α item agreement[48], which normalizes agreement by corpus for multi-corpora comparison, with Nominal, Ordinal, or Ratio distance metrics as appropriate. α agreement is expressed in the range (-1.0 – 1.0); 1.0 is perfect agreement.

For each training strategy (*Integrated*, etc), the *training* instances were changed by the strategy, but the *test* instances were unaffected. For the division of test instances into Hard and Easy Cases, the training instances were unaffected, but the test instances were filtered by α item agreement. Hard/Easy Case parameters were chosen (not learned) to divide the corpus by item agreement into roughly[49] equal portions, relative to the corpus, for post-hoc error analysis. It was necessary to use different parameters for the different corpora due to highly differing agreement rates between the corpora, which can be seen by the case distribution in Table 4.1.

Agreement Parameters Training strategies *HighAgree* and *VeryHigh* utilize agreement cutoff parameters that vary per corpus. These strategies are a discretized approximation of the

[46]See Japkowicz and Shah (2011) for usage description.

[47]We followed Yano et al. (2010) and Strapparava and Mihalcea (2007) in using *mean* as gold standard. Although another aggregation such as *median* might be more representative, such discussion is beyond the scope of this work.

[48]as implemented in the DKPro Statistics library (Meyer et al., 2014)

[49]based on the discrete item agreement distribution

		Hard Cases		Easy Cases		Total Cases
Corpus	**Task**	α	%	α	%	#
YANO2010	Biased Lang	<-.21	.15	>.20	.48	1041 sentences
CARP2009	Stemming	<-.50	.12	>.50	.54	6679 word pairs
PASCAL RTE-1	RTE	<0	.29	>.30	.26	800 sent. pairs
GIMBELANNO	POS Tagging	<0	.04	>.49	.75	14,439 tokens
SEMANNO	Affect Rec.	<0	.20	>.30	.49	100 headlines

Table 4.1: Case distributions of the datasets.

	Strategy cutoffs	
Task	**HighAgree**	**VeryHigh**
Biased Lang	>-.2	>.4
Stemming	>-.1	n.a.
RTE	>0	>.3
POS Tagging	⩾.2	⩾.5
Affect Rec.	>0	>.3

Table 4.2: Summary of training strategy cutoffs. See respective task sections for details.

gradual effect of filtering low agreement instances from the training data. For any given corpus, we could not use a cutoff value equal to no filtering, or that eliminated a class. If there were only two remaining cutoffs, we used these. If there were more candidate cutoff values, we trained and evaluated a classifier on a development set and chose the value for *HighAgree* that maximized Hard Case performance on the development set.

Note that although item agreement is in theory a continuum, in practice each corpus has discrete *levels*: the corpus consists of n groups of instances where all instances in the group have equal item agreement. For example, a corpus with 5 crowdsource labels per instance and a majority vote gold standard has a maximum of three item agreement levels per class: 3-vote agreement, 4-vote agreement, and 5-vote agreement. A corpus with expert labels or with a higher number of crowdsource labels will have more item agreement levels.

Evaluation of Corpus Size In addition to comparing performance of different training strategies, we evaluate the impact of corpus size on training strategy performance. Specifically, we address the questions: "Which training strategy performs best with different size corpora?" and "What is the added benefit of additional high-agreement training instances compared to additional generic training instances?" To address the first question, for each task, we provide a training data size/F_1-or-r performance curve, based on training set size *before* filtering, for both a filtering strategy and for the integrated baseline. To address the second question, for each task, we provide a training data size/F_1-or-r performance curve, based on training set size *after* filtering, for both a filtering strategy and for the integrated baseline.

4.4 Biased Language Detection

This task detects the use of bias in political text. The YANO2010 corpus (Yano et al., 2010)[50] consists of 1,041 sentences from American political blogs. For each sentence, five crowdsource annotators[51] chose a label *no bias*, *some bias*, and *very biased*. We follow Yano et al. (2010) in representing the amount of bias on an ordinal scale (1-3), where 1=*no bias*, 2=*some bias*, and 3=*very biased*.

We built a SVM regression system using all unigrams[52], to predict the numerical amount of bias. Evaluation is by 10-fold cross validation (*CV*); n-gram length and count (1,000 most frequent n-grams) were tuned using development sets taken from the training data of each CV round. Because there are no expert labels for YANO2010, we use Integrated labels as gold labels.

Item-specific agreement was calculated with ordinal distance function (Krippendorff, 1980). Hard Cases (161 instances) were defined as α item agreement <-0.21, and Easy Cases (499 instances) were defined as α item agreement >0.2. (We do not discuss the Middle Cases, which represent the middle part of the continuum between Hard Cases and Easy Cases.) Samples of YANO2010, along with item agreement, are available in Appendix A.

We use the following training strategies:

Integrated The average of the instance's crowdsource labels.

VeryHigh Filtered for agreement >0.4.

HighAgree Filtered for agreement >-0.2.

SoftLabel One training instance is generated for each label that a worker gave to an annotation instance, and weighted by how many times that label occurred with the annotation instance.

SLLimited SoftLabel, except that training instances with a label distance >1.0 from the original annotation instance label average are discarded.

4.4.1 Results

Table 4.3 compares the different strategies. Overall, there was no benefit to removing low agreement training instances (*VeryHigh*, .140; *HighAgree*, .231) or soft labeling (*SoftLabel*, .223; *SLLimited*, .235) compared to the aggregated label (*Integrated*, .236). However, there was a statistically significant (paired TTest, $p < 0.05$) improvement from low-agreement filtering for Hard Cases with *HighAgree* (*HighAgree*, .210 versus *Integrated*, .144).

Figure 4.3 compares the removal of low agreement instances, using a linear cutoff, with label averaging (*Integrated*). As expected, filtered performance for all case types eventually decreases as increased filtering removes too many training instances. Before this happens,

[50] Available at `https://sites.google.com/site/amtworkshop2010/data-1`

[51] 9% of instances received 10 labels.

[52] N-grams were found to outperform the list of strongly biased words in (Yano et al., 2010)

Training	All	Hard	Easy
Integrated	.236	.144	.221
VeryHigh	.140	.010	.158
HighAgree	.231	.210	.222
SoftLabel	.223	.131	.210
SLLimited	.235	.158	.208

Table 4.3: Biased Language: Pearson correlation results of training strategies on all data and Hard and Easy Cases.

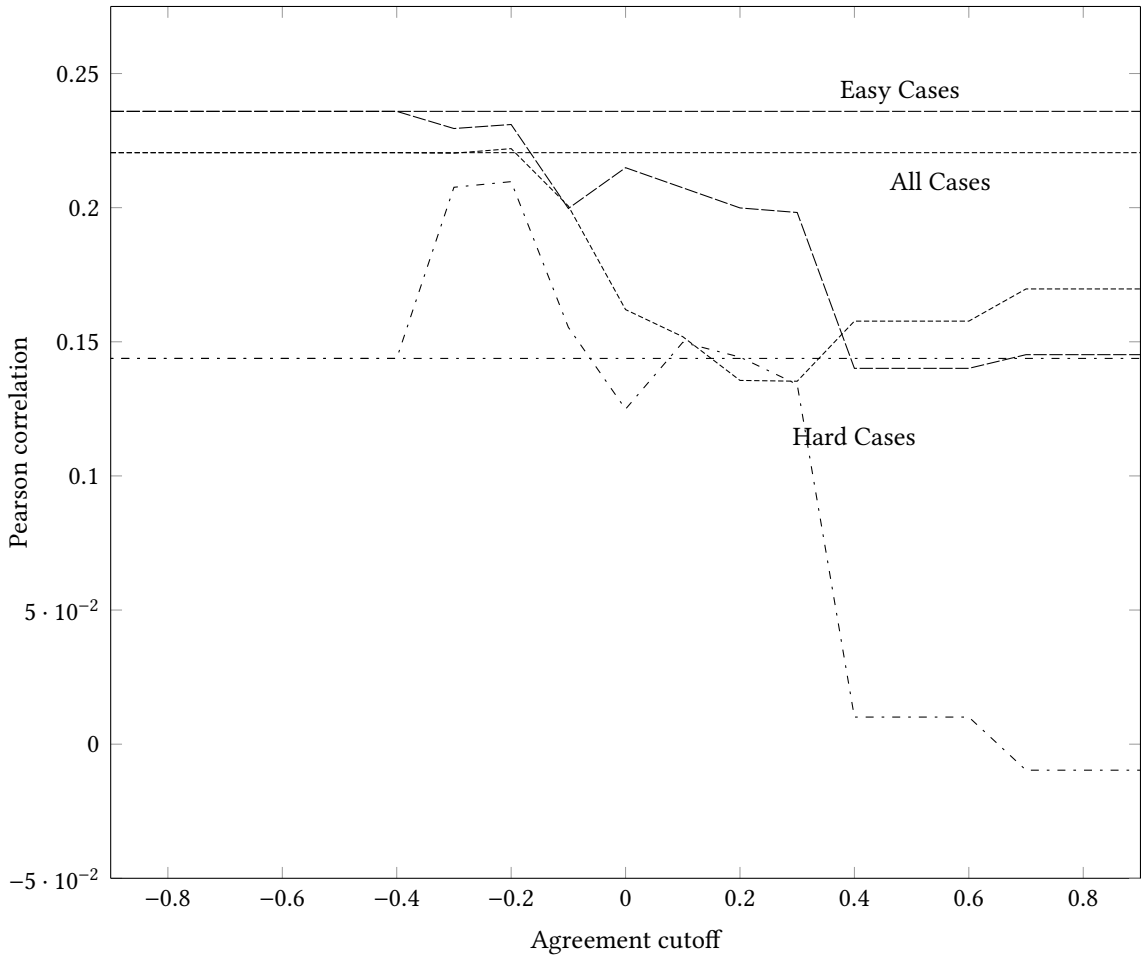

Figure 4.3: Biased Language: Filtering α *item agreement* cutoff curve, with Pearson correlation, for different case types (All, Hard, Easy); matching pattern lines show corresponding performance from the *Integrated* system.

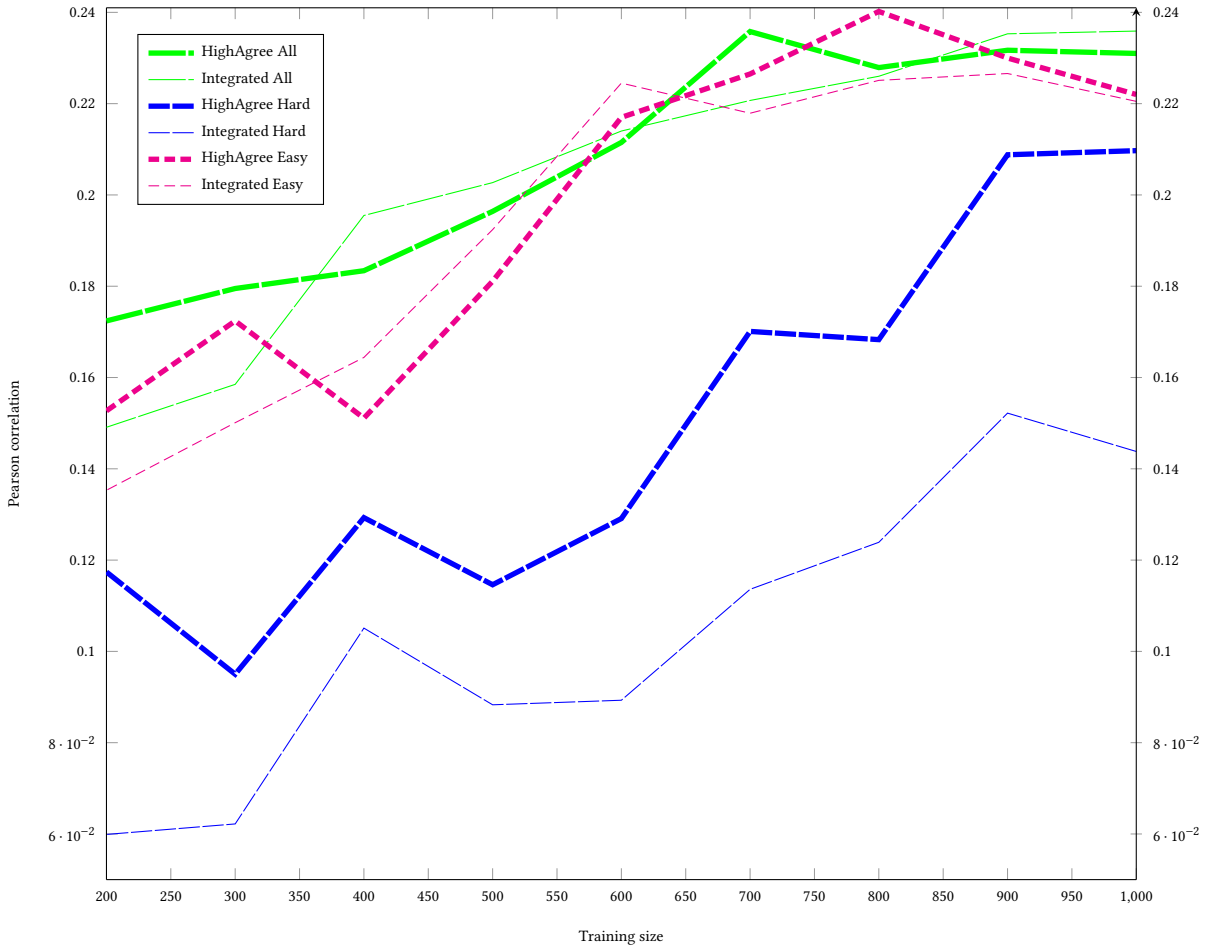

Figure 4.4: Biased Language: **before** filtering training size curve with different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems. *HighAgree* training size limit is 500.

however, there is a boost in Hard Case performance at α agreement cutoff=-0.2. We found similar filtering boosts in Hard Case performance in 4 of our 5 tasks.

Figure 4.4 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *before* filtering. For most cases, despite the smaller training data size, *HighAgree* performs similarly to *Integrated*, which shows that the system is not benefiting from the additional ambiguous instances in *Integrated*. However, for Hard Cases, this is particularly pronounced: Hard Case test instances are harder to classify when the training set included ambiguous instances. These findings suggest it may be useful to avoid ambiguous instances when crafting a dataset for annotation, no matter what the dataset size.

Figure 4.5 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *after* filtering. For all cases, given the same size training data, biased language detection is more accurate when there are

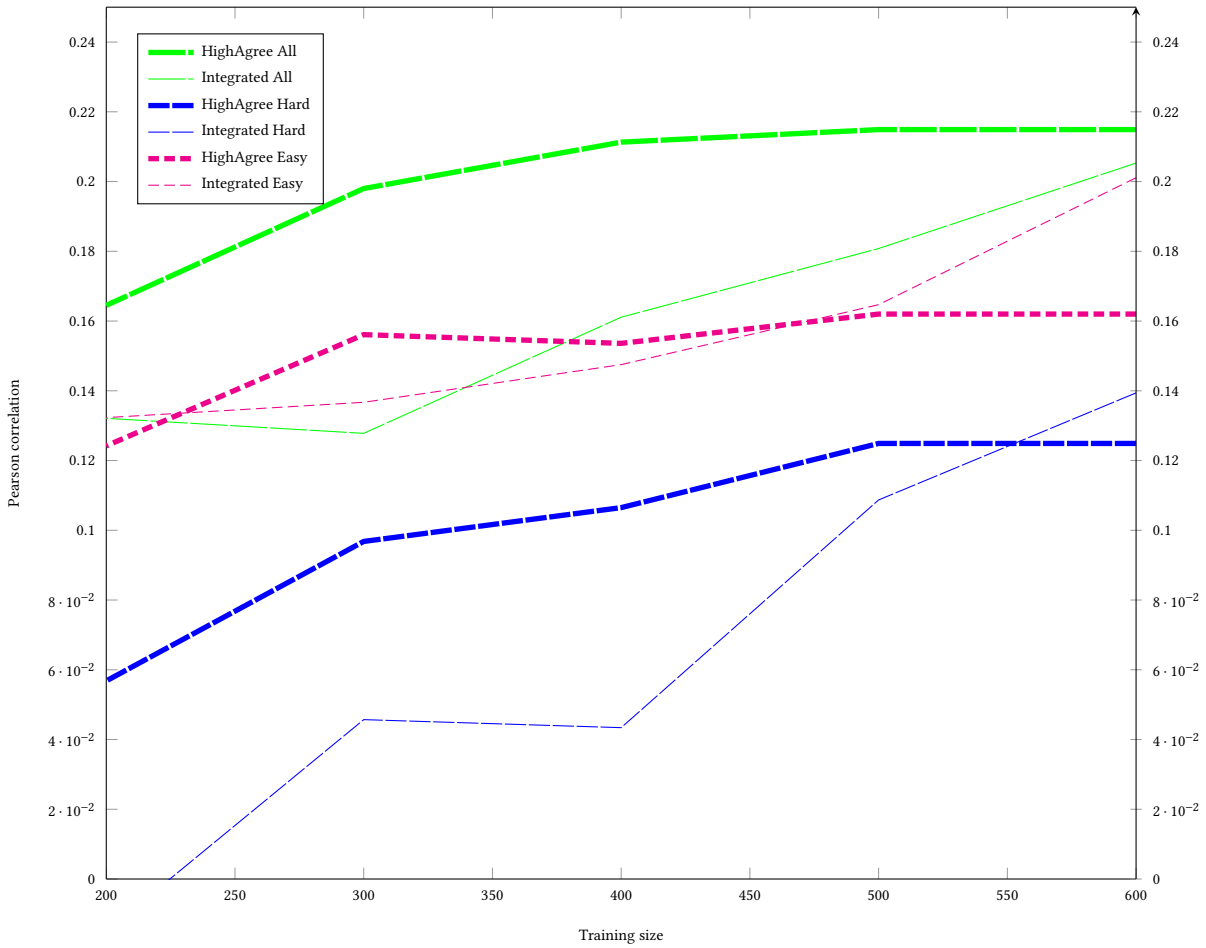

Figure 4.5: Biased Language: **after** filtering training size curve with different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems. *HighAgree* training size limit is 500.

Original	Stemmed	C	Original	Stemmed	C
box	box	+	instate	state	-
al	DELETED	+	goalkeeper	goalkeep	-
paused	pause	+	rockabilly	rock	-
workhorse	work horse	+	goalkeeper	DELETED	-

Table 4.4: Stems: Sample word pairs, with class C.

no low agreement instances in the training data; overall, *HighAgree* performs equivalently to *Integrated* with less training data. *HighAgree* instances are more valuable to the classifier than *Integrated* instances: a classifier trained on 300 *HighAgree* instances matches the performance of a classifier trained on over 500 *Integrated* instances. Increased performance per given training size is particular notable for Hard Cases. These findings suggest that preference should be given to high agreement instances in corpus construction.

4.5 Morphological Stemming

The goal of this binary classification task is to predict, given an original word and a stemmed version of the word, whether the stemmed version has been correctly stemmed. Specifically, the stemmed word should contain one less affix; or if the original word was a compound, the stemmed word should have a space inserted between the components; or if the original word was misspelled, the stemmed word should be deleted; or if the original word had no affixes and was not a compound and was not misspelled, then the stemmed word should have no changes. The CARP2009 dataset was compiled by Carpenter et al. (2009)[53]. Sample pairs are shown in Table 4.4.

The CARP2009 dataset contains 6679 word pairs. Labels were acquired from MTurk[54], and most pairs have 5 labels. We used all pairs. Our experiments used SVM and 10-fold CV, in which no pairs with the same original word could be split across training and test data. The gold standard was the *Integrated* label, with 4898 positive and 1781 negative pairs. Hard Cases (405 positive and 417 negative instances) were defined as CARP2009 instances with α item agreement <-0.5, and Easy Cases (3615 positive and 0 negative instances) were defined as CARP2009 instances with α item agreement >0.5. Samples of CARP2009, along with item agreement, are available in Appendix A.

Features used are combinations of the characters after the removal of the longest common substring between the word pair, including 0-2 additional characters from the substring; word

[53] Available at `https://github.com/bob-carpenter/anno`

[54] `mturk.com`

Word pair: rating, rate
Longest common substring = "rat"
Remainders = "ing" and "e"
Features = "_ingE_eE", "t_ingE_teE", "at_ingE_eE"

Figure 4.6: Feature creation for word pairs. Word boundary markers: B=beginning, E=end.

Training	All	Hard	Easy
Integrated	.797	.568	.927
HighAgree	.796	.569	.924
SoftLabel	.766	.539	.957
SLLimited	.799	.569	.930

Table 4.5: Stems: Micro-F_1 results of training strategies on all data and Hard and Easy Cases.

boundaries are marked. 1000 most-frequent-in-training strings were used.[55] An example is shown in Figure 4.6.

Training strategies include:

Integrated The most frequent (majority vote) label from the set of crowdsourced labels. Ties are decided randomly.

HighAgree Filtered for agreement >-0.1.

SoftLabel One training instance is generated for each label from a text, and weighted by how many times that label occurred with the text.

SLLimited *Integrated* with instances weighted by the frequency of the label among labels for the text pair.

4.5.1 Analysis

The results of our stemming classification are shown in Table 4.5. The dataset only had three levels of α item agreement, and the highest level had no negative instances. Therefore, it was not possible to examine the filtering strategy with a range of cutoffs.

The results in Table 4.5 show that there is no benefit to removing low agreement instances from the training set (*Integrated* .797 versus *HighAgree* .796), or to using soft labeling (*SoftLabel*, .766) or modified soft labeling (*SLLimited*, .799, improvement not significant), either overall or on Hard or Easy Cases. This was our only classifier that did not show Hard Case improvement with *HighAgree*. The Carp2009 corpus had the lowest number of item agreement levels among the five tasks, preventing fine-grained agreement training filtering, which explains why filtering shows no benefit.

[55] Character n-grams did not increase performance beyond the features mentioned, and were not used in final experiments.

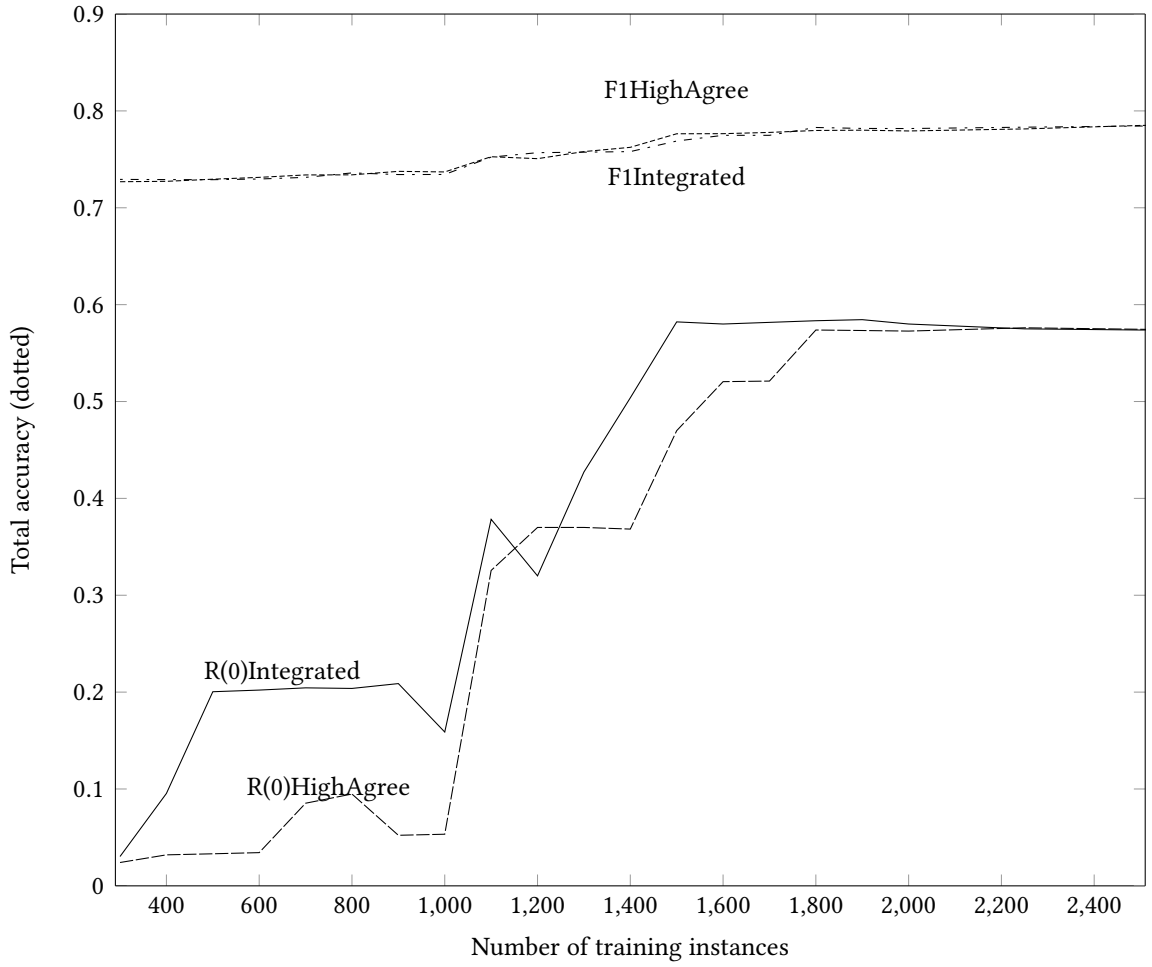

Figure 4.7: Stems: Training size curve and corresponding micro-F_1 and Recall(0) for *Integrated* versus *HighAgree*. The two micro-F_1 lines are tightly overlapped.

Text: *Cavern Club sessions paid the Beatles £15 evenings and £5 lunchtime.*
Hypothesis: *The Beatles perform at Cavern Club at lunchtime.*
Entailment Status: True

Figure 4.8: Sample text and hypothesis from RteAnno.

Figure 4.7 compares *Integrated* with *HighAgree* on micro-F_1 as well as negative recall ($\frac{TN}{TN+FP}$). Because the negative class is the minority class, an uninformative feature set with classifier merely outputting the class prior will result in negative recall near 0.

Figure 4.7 shows that *HighAgree* underperforms *Integrated* on negative recall. Carp2009 class distributions have different agreement rates: annotators agreed much more frequently on positive instances than on negative instances. In fact, there were no unanimous negative instances, but there were many unanimous positive instances. When low agreement training instances are removed, negative class instances are disproportionally removed; to compensate, a larger total training size is needed for the classifier to learn patterns of negative instances.

4.6 Recognizing Textual Entailment

Recognizing textual entailment is the process of determining whether or not, given two sentences (text + hypothesis), the meaning of one sentence can be inferred from the other. An example text-hypothesis pair is shown in Figure 4.8.

We used the RteAnno dataset from Pascal Rte-1 (dataset of the PASCAL Recognizing Textual Entailment Challenge (Dagan et al., 2006)), which contains 800 sentence pairs and annotations by trained annotators. The crowdsource annotations of 10 labels per pair were obtained by Snow et al. (2008)[56] from MTurk. Hard Cases (230 instances) were defined as α item agreement <0.0, and Easy Cases (207 instances) were defined as α item agreement >0.3. Samples of Pascal Rte-1 and RteAnno, along with item agreement, are available in Appendix A.

We reproduced the basic system described in (Dagan et al., 2006) of tf-idf weighted cosine similarity between lemmas of the text and hypothesis. The weight of each word_i in $document_j$, with N total documents, is the log-plus-one term_i frequency normalized by raw term_i document frequency, with Euclidean normalization.

$$\text{weight}(i,j) = \begin{cases} (1 + \log(\text{tf}_{i,j}))\frac{N}{\text{df}_i} & \text{if } \text{tf}_{i,j} \geqslant 1 \\ 0 & \text{if } \text{tf}_{i,j} = 0 \end{cases}$$

Additionally, we use features including the difference in noun chunk character and token length, the difference in number of tokens, shared named entities, and the RTE subtask

[56] Available at `https://sites.google.com/site/nlpannotations/`

Training	All	Hard	Easy
Integrated	.513	.330	.831
VeryHigh	.499	.304	.836
HighAgree	.543	.361	.831
SoftLabel	.499	.304	.836
SLLimited	.493	.291	.831

Table 4.6: RTE: Micro-F_1 results of training strategies on all data and Hard and Easy Cases.

names. We use the original labels from trained annotators from Dagan et al. (2006) as our gold standard. Training strategies are from Biased Language (*VeryHigh*) and Stem (*Integrated*, *HighAgree*, *SoftLabel*, and *SLLimited*) experiments, with the parameters of *HighAgree* cutoff as 0.0 and *VeryHigh* cutoff as 0.3. Experiments used SVM and 10-fold CV.

4.6.1 Results

The results of different training strategies are shown in Table 4.6. Removing low agreement training instances has a statistically significant (McNemar's Test (McNemar, 1947), $p < 0.05$) beneficial effect (*Integrated*, .513; *HighAgree*, .543). Figure 4.9 shows the filtering benefit, which occurs around α agreement cutoff=0. Neither soft labeling strategy (*SoftLabel* nor *SLLimited*) proved helpful: either the label noise in low agreement instances was not due to linguistic ambiguity and there was nothing to be learned via soft labeling, or soft labeling was not an effective mechanism to convey linguistic ambiguity to the machine learner.

Dagan et al. (2006) report performance of this system, on a different data division, of accuracy[57]=0.568, which is similar to our results.

Figure 4.10 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *before* filtering. For Easy Cases, despite the smaller training data size, *HighAgree* steadily outperforms *Integrated* across different training sizes. For Hard Cases, and in contrast to the Biased Language task, the reverse is true: *Integrated* outperforms *HighAgree*. For all cases, the crossover between *HighAgree* filtering too much training data versus learning a better model from high agreement training instances appears to occur at a training size of about 600 instances.

Figure 4.11 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *after* filtering. For Easy Cases and All Cases, given the same size training data, RTE is more accurate when there are no low agreement instances in the training data; overall, *HighAgree* performs equivalently to *Integrated* with less training data. *HighAgree* instances are more valuable to the classifier than Integrated instances: a classifier trained on 450 *HighAgree* instances matches the performance

[57]While we report our nominal class experiment results in micro F_1 for consistency across our three nominal class tasks, for the class-balanced binary RTE RTEANNO dataset, micro F_1 is equivalent to accuracy.

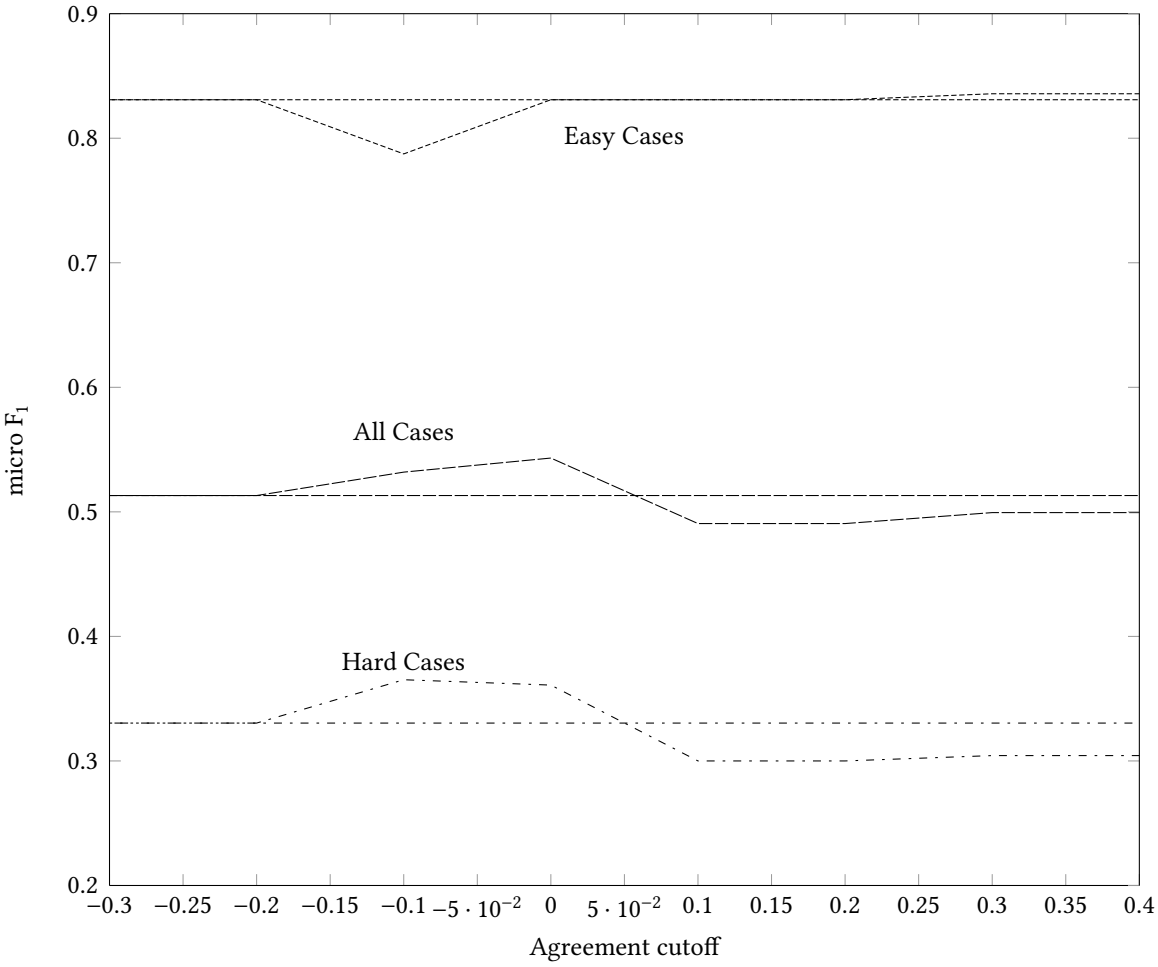

Figure 4.9: RTE: Filtering α *item agreement* cutoff curve, with micro-F_1, for different case types (All, Hard, Easy); matching pattern lines show corresponding performance from the *Integrated* system.

of a classifier trained on about 800 Integrated instances. In contrast, for Hard Case evaluation, it is unclear whether additional Integrated or *HighAgree* instances are more valuable. But for All Cases and Easy Cases,the findings suggest preference should be given to high agreement instances in corpus construction.

4.7 POS tagging

In this section, we build a POS-tagger for Twitter posts. We used the training section of the GIMBEL2011 dataset from Gimpel et al. (2011). Crowdsource labels for this data came from the GIMBELANNO dataset[58] (Hovy et al., 2014); there were 5 labels for each Tweet. After aligning and cleaning the crowdsourced corpus and the original corpus, our dataset consisted of 953

[58] Available at `http://lowlands.ku.dk/results/`

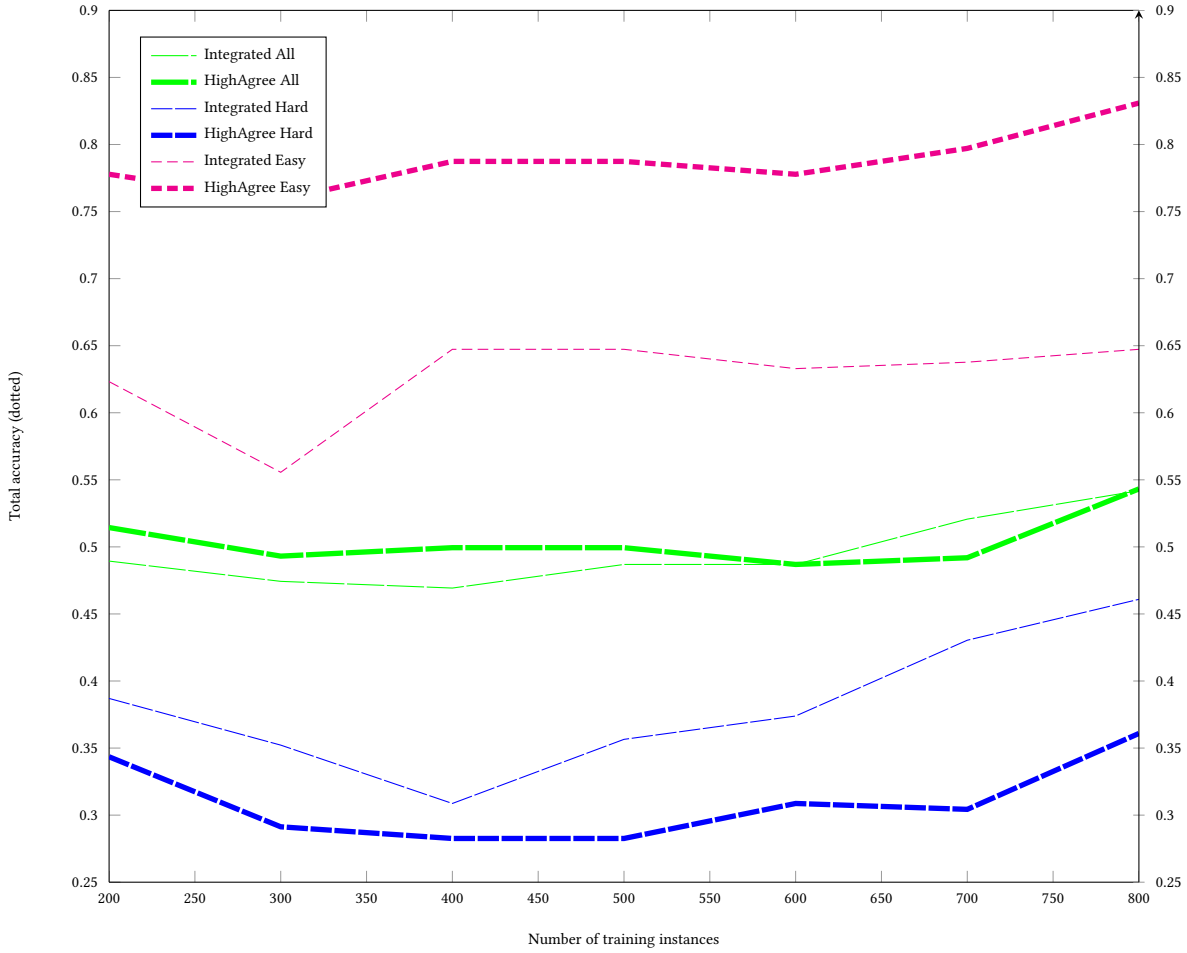

Figure 4.10: RTE: training size curve of micro-F_1 with different case types (All, Hard, Easy) for *Integrated* versus *HighAgree*. Size determined **before** filtering.

tweets of 14,439 tokens. Hard Cases (649 instances) were defined as α item agreement <0.0, and Easy Cases (10830 instances) were defined as α item agreement >0.49. Samples of GIMBEL2011 and GIMBELANNO, along with item agreement, are available in Appendix A.

We followed Hovy et al. (2014) in constructing a CRF classifier (Lafferty et al., 2001), using a list of English affixes, Hovy et al. (2014)'s set of simple orthographic features, and word clusters (Owoputi et al., 2013). We evaluated by means of 5-fold CV[59], in which individual tweets were assigned to folds. The POS tagset was the universal tag set (Petrov et al., 2012); we converted Gimpel et al. (2011)'s tags to the universal tagset (12 tags) using Hovy et al. (2014)'s mapping. The gold standard was the *Integrated* label. A sample Tweet and its crowdsourced labels are shown in Table 4.7.

We used the following strategies:

Integrated Majority vote.

[59]POS Tagging experiments were computationally intensive, so 5-fold CV was used instead of 10-fold.

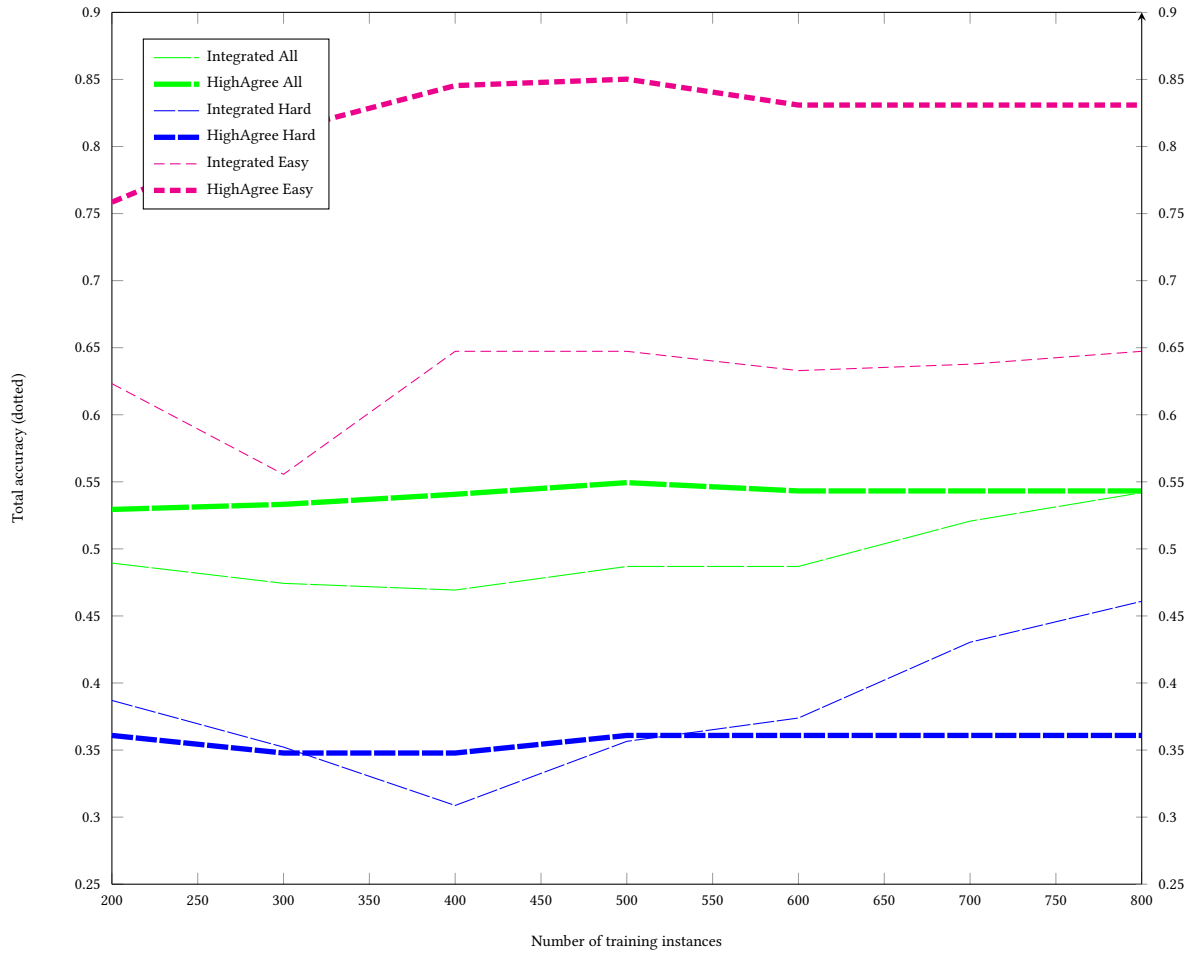

Figure 4.11: RTE: training size curve of micro-F_1 with different case types (All, Hard, Easy) for *Integrated* versus *HighAgree*. Size determined **after** filtering.

HighAgree For each token t in sequence s (i.e., a tweet) where α-$agreement(t)$ <0.2, s is broken into two separate sequences s_1 and s_2 and t is deleted. Illustrated in Figure 4.12.

VeryHigh *HighAgree* with agreement <0.5.

SoftLabel For each proto-sequence s (i.e., a tweet), we generate 5 sequences $\{s_0, s_1, ..., s_i\}$, in which each token t is assigned a crowdsource label drawn at random: $l_{t,s_i} \in L_t$.

SLLimited Each token t in sequence s is assigned its *Integrated* label. Then s is given a weight representing the average item agreement for all $t \in s$.

4.7.1 Results

The results of the different POS-tagging training strategies are shown in Table 4.8. Hovy et al. (2014) reported Majority Vote results (acc=.805 – .837 on a different data division) similar to our *Integrated* results of .790 micro-F_1. The best performing strategy was *HighAgree*: it sig-

Token	Crowdsourced labels
@USER	NOUN, NOUN, NOUN, NOUN, NOUN
I	PRON, PRON, PRON, PRON, X
like	VERB, ADJ, VERB, VERB, VERB
monkeys	NOUN, NOUN, NOUN, NOUN, NOUN
,	., ., ., NOUN, .
but	CONJ, CONJ, CONJ, CONJ, CONJ
I	PRON, PRON, PRON, PRON, PRON
still	ADV, ADV, ADV, ADV, ADV
hate	VERB, VERB, VERB, VERB, VERB
Costco	NOUN, NOUN, NOUN, X, NOUN
parking	NOUN, ADJ, VERB, NOUN, ADJ
lots	NOUN, NOUN, NOUN, NOUN, NOUN
..	X, ., ., ., .

Table 4.7: POS Tags: A sample Tweet and labels.

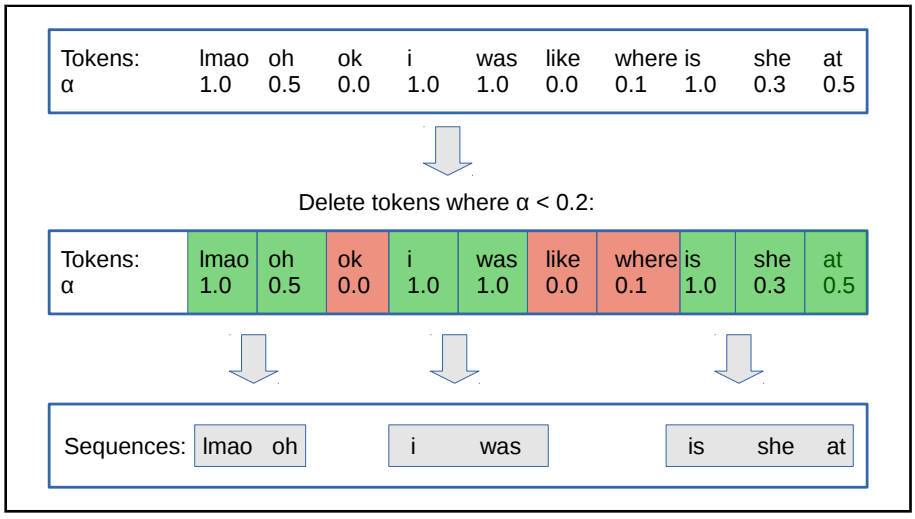

Figure 4.12: Illustration of *HighAgree* (cutoff = 0.2) for POS tagging.

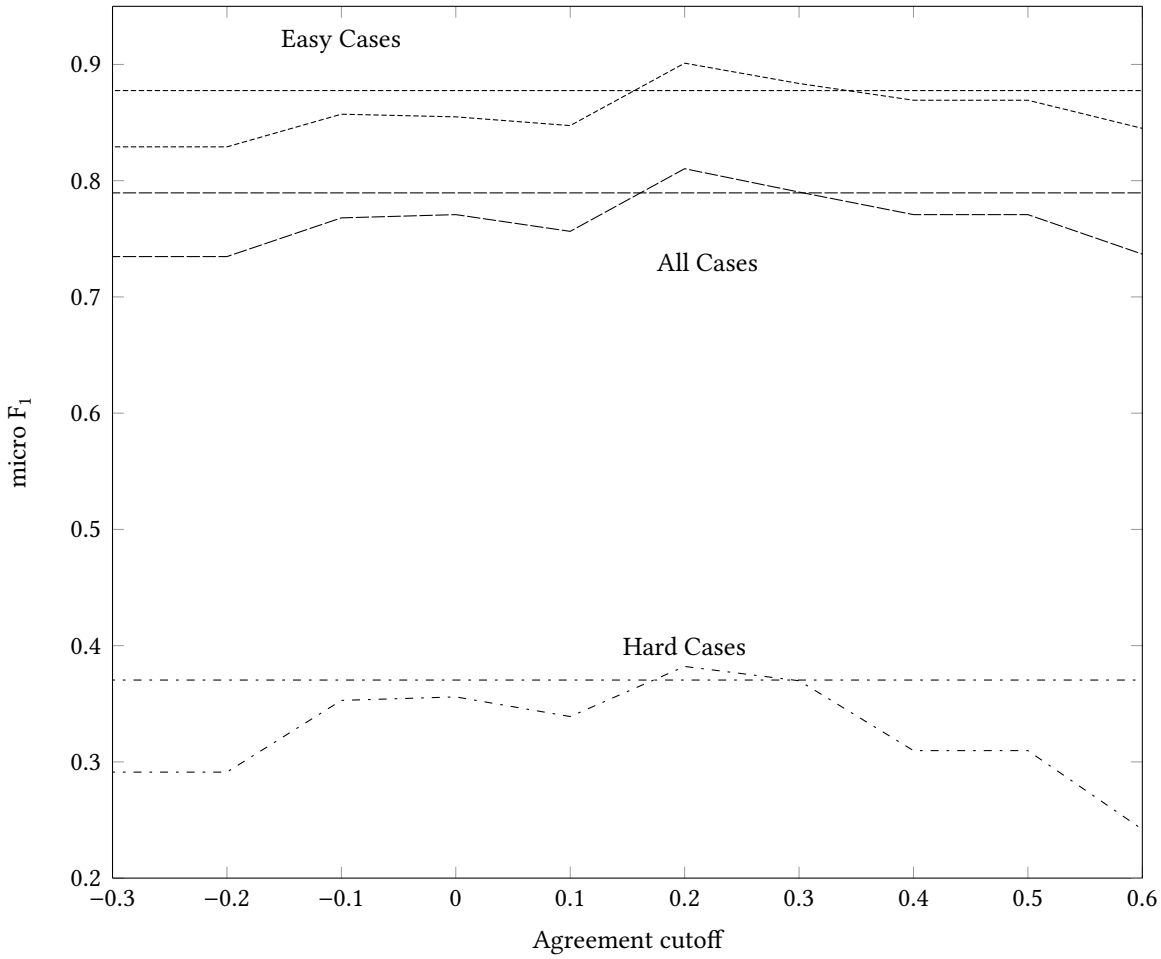

Figure 4.13: POS Tags: Filtering α *item agreement* cutoff curve, with micro-F_1, for different case types (All, Hard, Easy); matching pattern lines show corresponding performance from the *Integrated* system.

nificantly outperformed all other strategies overall as well as in each of the instance difficulty categories (McNemar's Test, $p < 0.05$). Regarding the soft labeling strategies, where neither *SoftLabel* nor *SLLimited* were significantly helpful, we draw the same conclusion as with the biased language task and RTE task: either the label noise in low agreement instances was not due to linguistic ambiguity and there was nothing to be learned via soft labeling, or soft labeling was not an effective mechanism to convey linguistic ambiguity to the machine learner.

The benefit of *HighAgree* can be seen in Figure 4.13, when the curves of all cases outperform *Integrated* at α agreement cutoff=0.2.

Figure 4.14 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *before* filtering. The lower training sizes from *HighAgree* filtering show a detrimental effect in comparison with *Integrated* on All Cases, Easy Cases, and Hard Cases, until training size reaches about 12,000

Training	All	Hard	Easy
Integrated	.790	.370	.878
VeryHigh	.771	.310	.869
HighAgree	.810	.382	.901
SoftLabel	.789	.353	.880
SLLimited	.797	.376	.880

Table 4.8: POS Tags: Micro-F_1 results of training strategies on all data and Hard and Easy Cases.

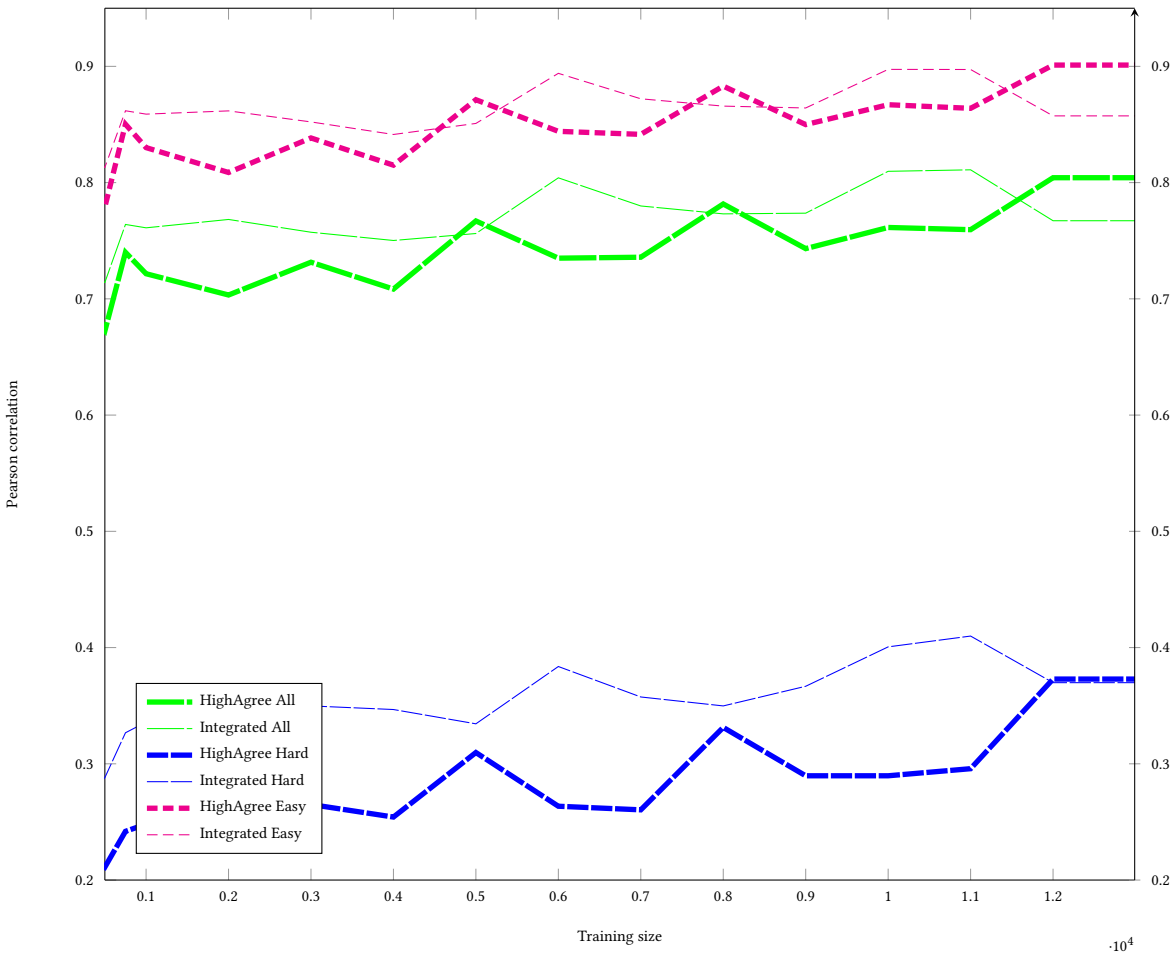

Figure 4.14: POS Tags: training size curve of micro-F_1 with different case types (All, Hard, Easy) for *Integrated* and *HighAgree* systems. Training size **before** filtering.

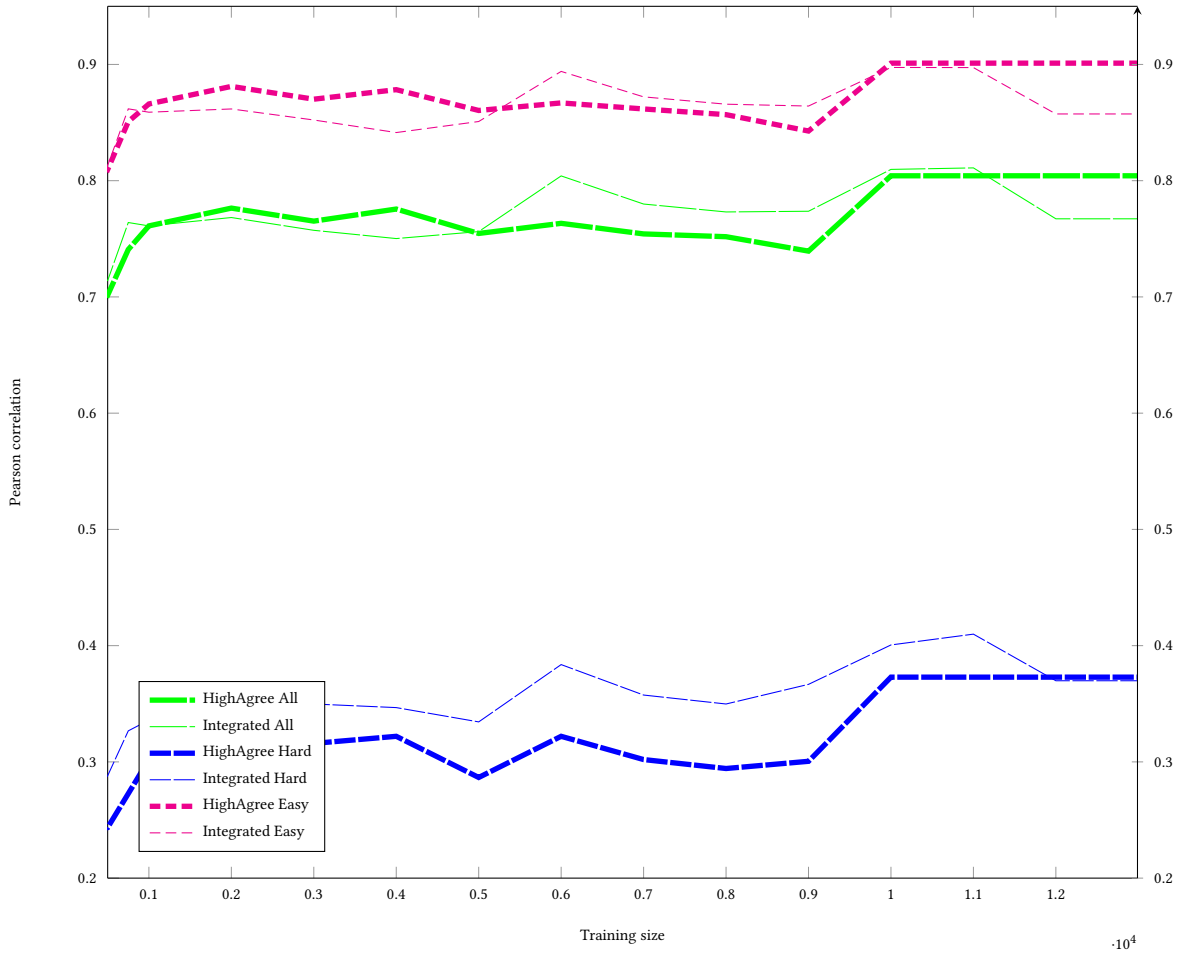

Figure 4.15: POS Tags: training size curve of micro-F_1 with different case types (All, Hard, Easy) for *Integrated* and *HighAgree* systems. Training size **after** filtering.

tokens. (It is not clear if the best performance switch from *Integrated* to *HighAgree* at 12,000 tokens is the final switch, because the final training size is less than 13,000.)

Figure 4.15 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *after* filtering.

In contrast to the Biased Language and RTE tasks, the value of an additional *HighAgree* instance does not always outweigh the value of an additional *Integrated* instance, for learning a better model. When training size is small (less than 5000 tokens), *HighAgree* instances are slightly more valuable to the classifier than *Integrated* instances when evaluating on All Cases or Easy Cases; as training size increases, this fluctuates between *HighAgree* and *Integrated* instances. But when evaluating on Hard Cases, an additional *Integrated* instance is almost always more valuable than an additional *HighAgree* instance. This may be due to uneven downsampling between classes with different agreement rates (and the POS tagging task has 12 classes, the highest of any of our nominal class tasks), causing some classes to be under-represented in *HighAgree* instances. In this situation, even a noisy low agreement

instance may be more valuable to the classifier than a *HighAgree* instance, because it comes from an underrepresented class.

HighAgree instances become universally more valuable than Integrated instances when training size reaches about 12,000 tokens, at which point the classifier may have a sufficient number of training instances of each class, that it prefers additional high agreement instances over additional lower agreement instances.

4.8 Affect Recognition

Our Affect Recognition experiments are based on the affective text annotation task in Strapparava and Mihalcea (2007). In the original community task, news headlines are rated for six emotions: anger, disgust, fear, joy, sadness, and surprise; we evaluate only the *sadness* Sem2007 dataset, which had the highest inter-annotator agreement among trained annotators, r=68.19 (Strapparava and Mihalcea, 2007). Each headline is rated for "sadness" using a scale of 0-100. The goal of affect recognition is to model the connection between emotions and lexical semantics. An example is in Figure 4.16. We use the SemAnno dataset, a crowdsourced annotation for a 100-headline sample of Sem2007, provided by Snow et al. (2008)[60], with 10 annotations per emotion per headline. Hard Cases (20 headlines) were defined as α item agreement <0.0, and Easy Cases (48 headlines) were defined as α item agreement >0.3. (As previously explained in Section 4.3, Case parameters were chosen for most-equal corpus division, rather than learned on a development set.) Gold labels were created by trained annotators (Strapparava and Mihalcea, 2007). Samples of Sem2007 and SemAnno, along with item agreement, are available in Appendix A.

Our system design is identical to Snow et al. (2008), which is similar to the SWAT system (Katz et al., 2007), a top-performing system on the SemEval Affective Text task, except that for unseen tokens, we assign a value equal to the average emotion score (Snow et al. do not specify how they handled unseen tokens). We use 10-fold CV.

> From Snow et al. (2008):
> For each token `t` in our training set, we assign t a weight for each emotion e equal to the average emotion score observed in each headline H that t participates in. i.e., if $\mathbf{H}_t$ is the set of headlines containing the token t, then:
>
> $$Score(e, t) = \frac{\sum_{H \in H_t} Score(e, H)}{H_t}$$
>
> With these weights of the individual tokens we may then compute the score for an emotion e of a news headline H as the average score over the set of tokens $t \in H$

[60] Available at `https://sites.google.com/site/nlpannotations/`

Headline: *Hussein's niece pleads for father's life*
'Sadness' ratings: 10, 100, 0, 0, 0, 70, 100, 20, 100, 100

Figure 4.16: Affective text example.

Training	All	Hard	Easy
Integrated	.446	.115	.476
VeryHigh	.326	.059	.376
HighAgree	.453	.265	.505
SoftLabel	.450	.112	.477
SLLimited	.450	.139	.472

Table 4.9: Affective Text: Pearson correlation results of training strategies on all data and Hard and Easy Cases.

that we've observed in the training set, i.e.,

$$Score(e, H) = \sum_{t \in H} \frac{Score(e, t)}{|H|}$$

Where $|H|$ is simply the number of tokens in headline H, ignoring tokens not observed in the training set.

Training strategies are similar as for the Biased Language experiments, except:
VeryHigh Filtered for agreement >0.3.
HighAgree Filtered for agreement >0.
SoftLabel Same as Biased Language task: one training instance is generated for each label of a headline, and weighted by how many times that label occurred with the headline.
SLLimited Like SoftLabel, except that instances with a label distance >20.0 from the original label average (i.e., if the original label average is 57 and a single label is 12, then the label distance is 45) are discarded.

4.8.1 Results

Results on the *Sadness* dataset are shown in Table 4.9. *HighAgree* outperformed *Integrated* by a small but statistically significant (paired TTest, $p < 0.05$) margin (*HighAgree*, .453; *Integrated*, .446). Improvement was much larger for Hard Cases, where the lack of ambiguous training instances led to a 15 percentage point improvement. The benefit of *HighAgree* can be seen in Figure 4.17, with maximal performance at α agreement cutoff=0.1. Earlier, we saw similar improvement for Hard Cases in the Biased Language, RTE, and POS Tagging experiments.

Figure 4.18 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *before* filtering. The lower

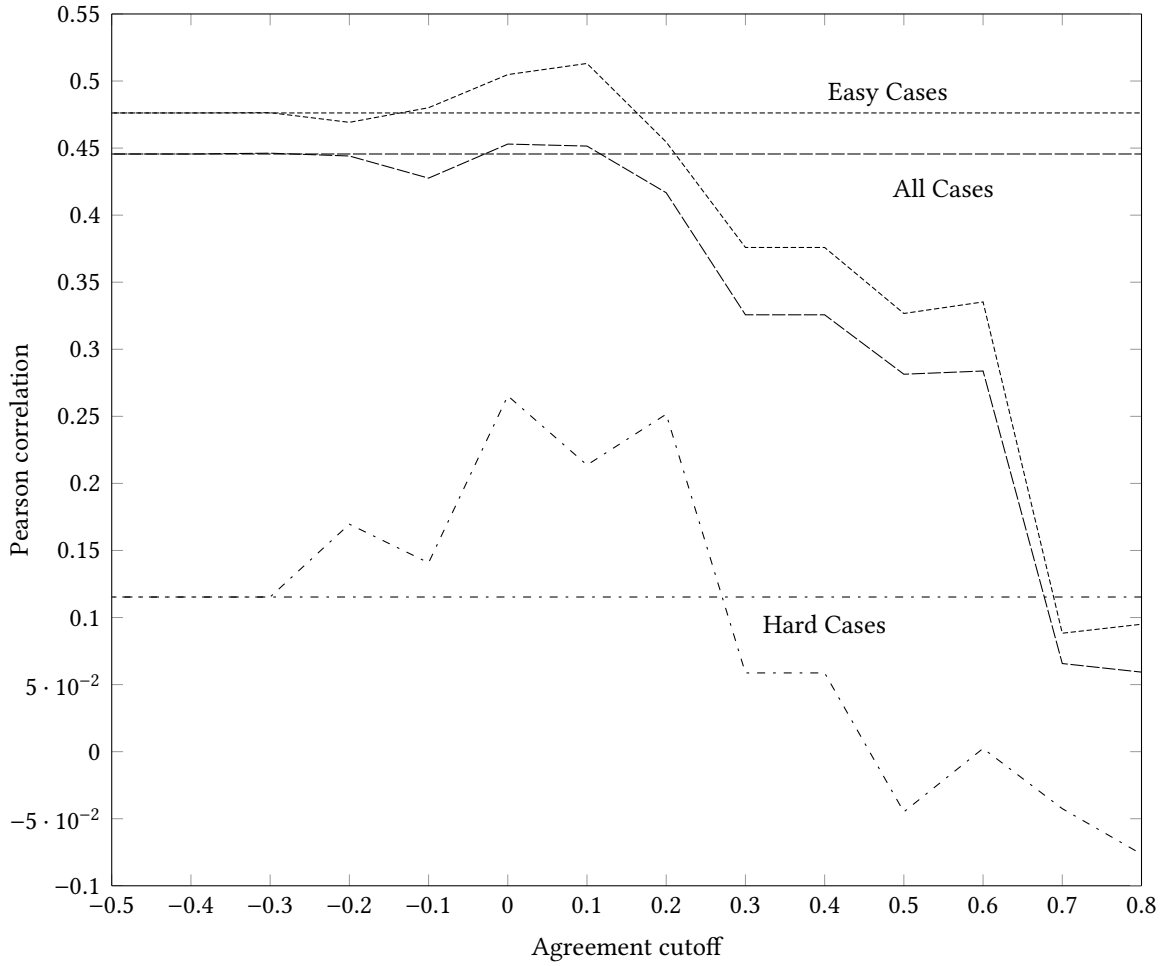

Figure 4.17: Affective Text: Filtering α *item agreement* cutoff curve, with Pearson correlation, for different case types (All, Hard, Easy); matching pattern lines show corresponding performance from the *Integrated* system.

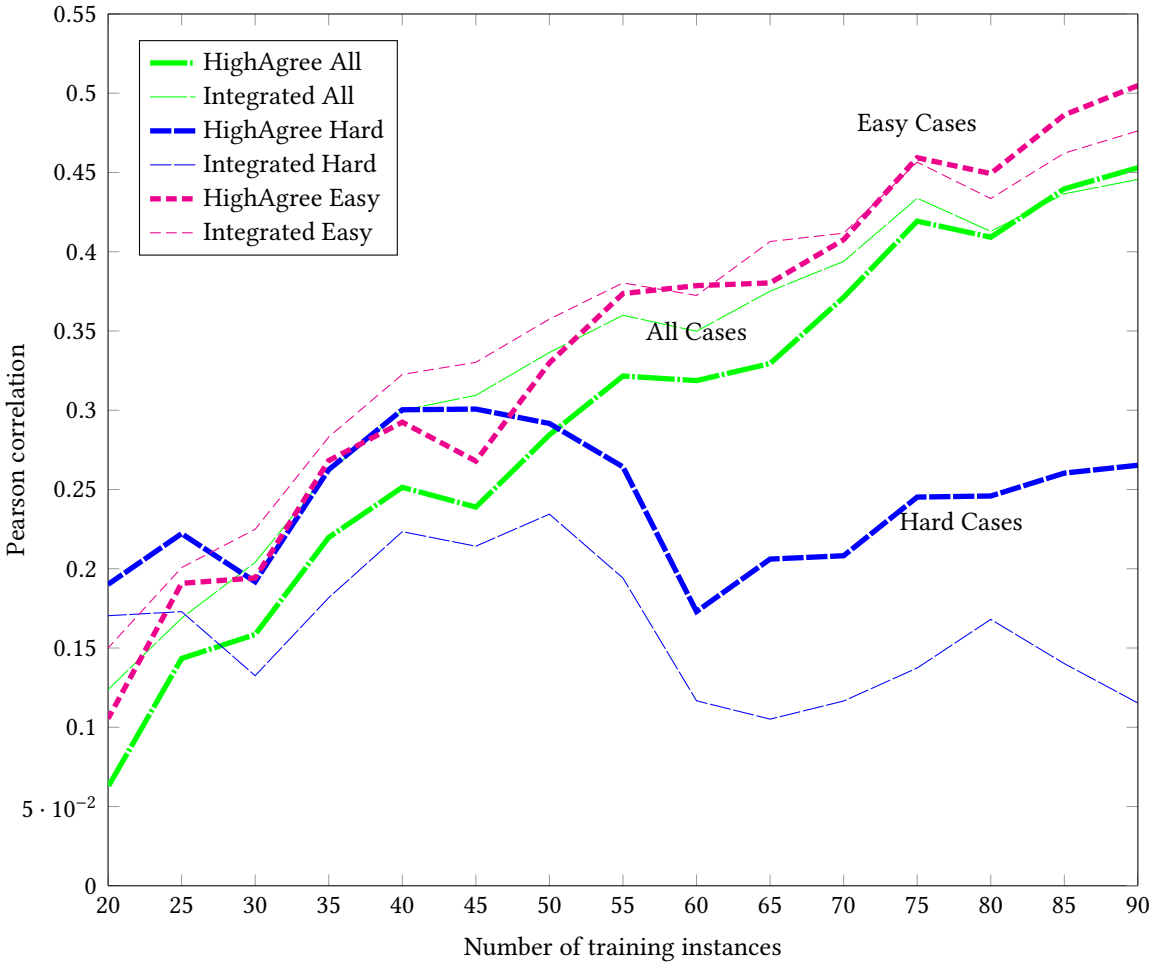

Figure 4.18: Affective text: Training size **before** filtering, with Pearson correlation and for different case types (All, Hard, Easy); similar pattern lines with single dots show corresponding performance from the *Integrated* system. Averaged over 5 runs of 10-fold CV, or 10 runs for Hard Cases.

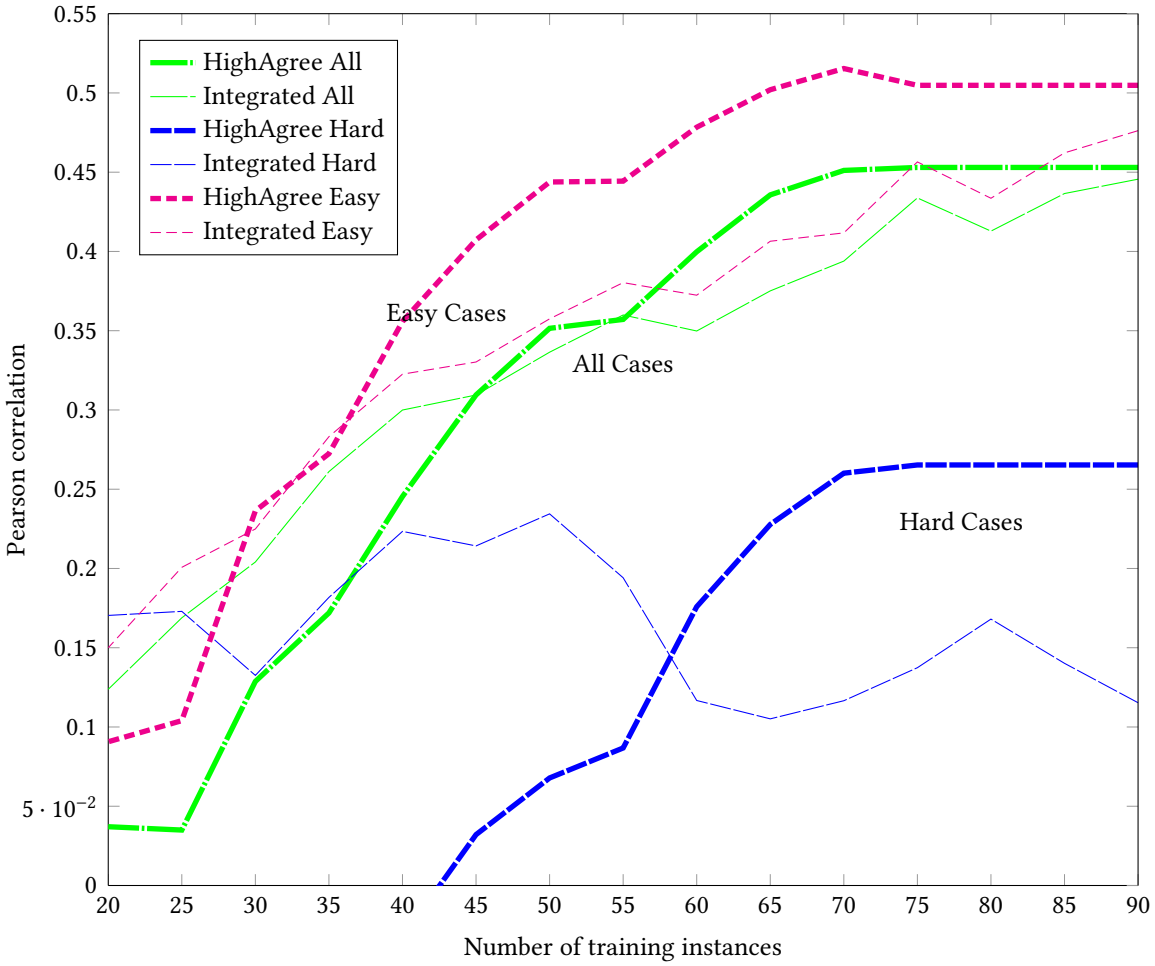

Figure 4.19: Affective text: Training size **after** filtering, with Pearson correlation and for different case types (All, Hard, Easy); similar pattern lines with single dots show corresponding performance from the *Integrated* system. Averaged over 5 runs of 10-fold CV, or 10 runs for *Integrated* Hard Cases.

training sizes from *HighAgree* filtering show a detrimental effect in comparison with *Integrated* on All Cases and Easy Cases, until training size reaches about 75 headlines. However, when evaluating on Hard Cases, *HighAgree* consistently outperforms *Integrated* across all training sizes.

Figure 4.19 shows the training size curve for different case types (All, Hard, Easy), for *Integrated* and *HighAgree* systems, when training size is calculated *after* filtering. With larger training sizes, *HighAgree* instances are universally more valuable. But similarly to the POS tagging task, the value of an additional *HighAgree* instance does not always outweigh the value of an additional *Integrated* instance, for learning a better model. When training size is small, *HighAgree* instances are slightly more valuable to the classifier than *Integrated* instances. This crossover point ranges from 30 headlines for Easy Cases, 45 headlines for All Cases, to about 58 cases for Hard Cases. This may be due to uneven downsampling between classes with different

agreement rates, causing some classes to be under-represented in *HighAgree* instances. In this situation, even a noisy low agreement instance may be more valuable to the classifier than a *HighAgree* instance, because it comes from an underrepresented class.

We examined the SemAnno dataset for evidence of uneven downsampling between classes. We observe that annotators appeared to be representing two pieces of information with the same label: "Does this headline express this emotion?" and "How much does this headline express this emotion?" Annotators frequently agreed when a headline did not express sadness (i.e., a labelset of almost all "0"'s), but disagreed when it did express sadness. Such disagreement can be seen in Figure 4.16, where ratings range from 0 to 100, resulting in low item agreement. Removing or reducing weight for low agreement training instances effectually removes instances of the *sadness* class altogether. To compensate for the annotation bias towards "0"'s, we tried removing different amounts of zero-label annotations (i.e., labels of "0" sadness) and replacing them with the average value of the non-zero labels, before applying our training strategies. The goal of this change was to shift the focus from the annotation question "Does this headline express this emotion?", to the question "How much does this headline express this emotion?"

Figure 4.20 shows the results of removing these zero-label annotations by number of "0"'s removed (0-9). To avoid raising scores of instances that were true negatives (no sadness), we only replaced the zero-labels of an instance's labels if there were n or fewer zero-labels in the labelset. An n=4 results in a set of labels with no more than 4 zeros having its zero's replaced by the non-zero average. The peak improvement around n=2 shows that multiple meanings of a zero-label (i.e., "headline isn't sad" and "headline has a very low level of sadness") reduced system performance.

Snow et al. (2008) report results on a different data division of r=.174, a merged result from systems trained on combinations of crowdsource labels and evaluated against expert-trained systems. The SWAT system (Katz et al., 2007), which also used lexical resources and additional training data, achieved r=.3898 on a different section of data. These results are comparable with ours, which range r=.326 − .453.

4.9 Chapter Summary

In this chapter, we have investigated different strategies to train a machine classifier on corpora with noisy crowdsourced labels. For five natural language tasks, we have examined the impact of informing the classifier of item agreement, by means of soft labeling and low-agreement training instance filtering.

We answered the following questions, posed at the beginning of this chapter:

> *Research Question*: In the context of crowdsourced datasets, is the best classifier produced from a training dataset of integrated labels, or from item agreement filtering, or from soft labeling?

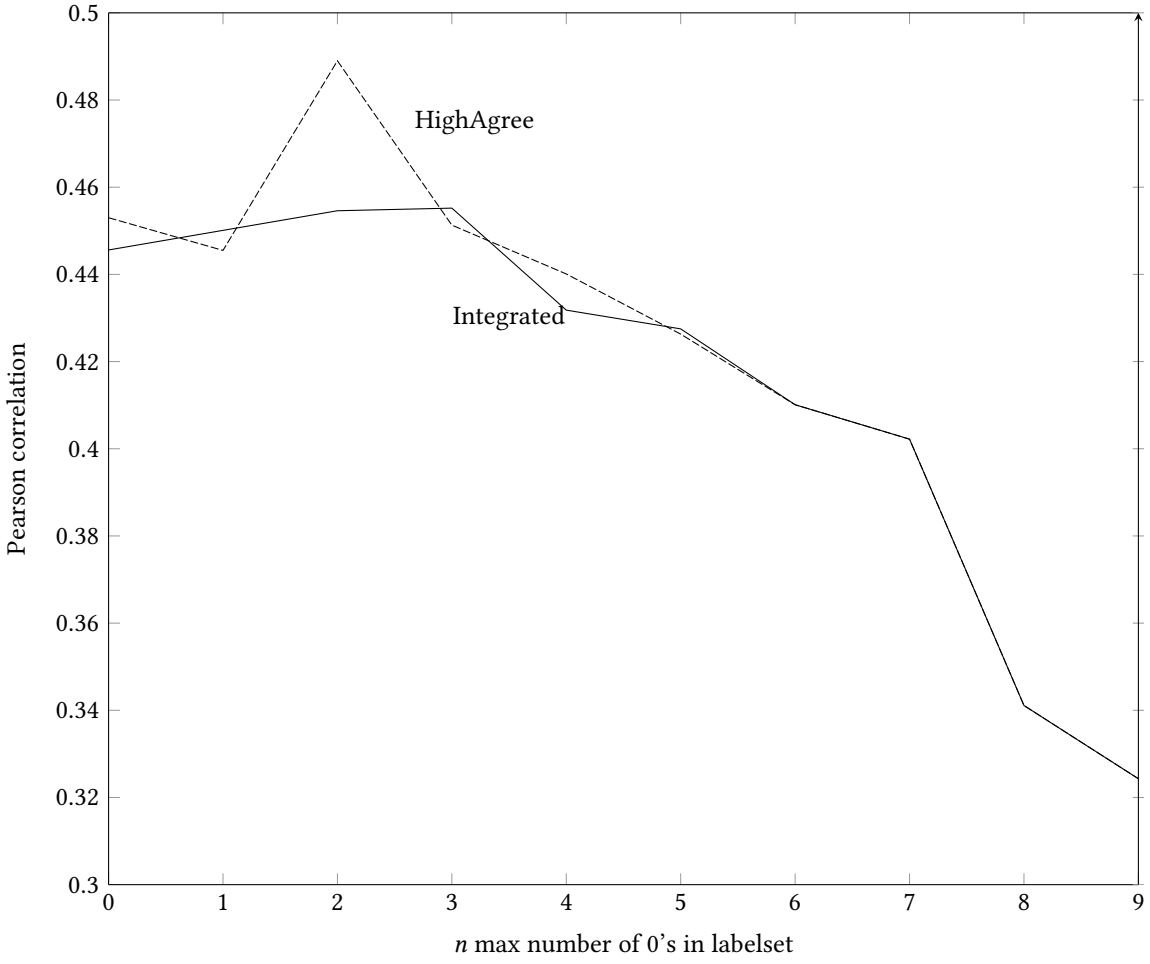

Figure 4.20: Affective Text: Pearson correlation result for *Integrated* and *HighAgree* when replacing zero-labels. X-axis shows the maximum number of zero-labels an instance's labelset can have such that the zero-labels are replaced with the average of the non-zero labels.

In four out of the five natural language tasks, we found a statistically significant benefit from filtering, compared to integrated labels. The fifth task, Stemming, had the lowest number of item agreement levels of the five tasks, preventing fine-grained parameter tuning of agreement filtering levels, which explains why filtering shows no benefit. Sheng et al. (2008) had previously suggested a theoretical approach to preserving crowdsourcing label uncertainty through a soft labeling approach. However, they do not apply this to a real dataset. We applied soft labeling techniques to five datasets, and we found no systematic performance benefit over an integrated baseline.

> *Research Question*: Some instances are naturally more difficult for humans to annotate. Do our strategies, as listed in the previous question, have an equal impact on both Hard Cases and Easy Cases in the test data?

Filtering, the best-performing strategy, showed strongest improvements on Hard Cases. The classifiers were not able to learn from the disagreement of the annotators, and this showed most clearly for borderline instances, the Hard Cases. However, we also observed our training strategies impact some classes more than others when the classes have unequal item agreement, increasing sample selection bias, which negatively impacts model learning. Our findings suggest that the best crowdsource label training strategy is to remove low item agreement instances, although care must be taken to account for different agreement rates per class so that a class with low agreement is not too undersampled.

> *Research Question*: How does corpus size impact performance of training strategy: Which training strategy performs best with different size corpora? What is the added benefit of additional high-agreement training instances compared to additional generic training instances?

We found that a *HighAgree* filtered training strategy produces a model equal or better than an *Integrated* (unfiltered) dataset of the same size as the *HighAgree* dataset before filtering, unless the training size is so small that *HighAgree* is underrepresenting particular classes due to over-downsampling of low agreement classes. This *HighAgree* good performance indicates that the machine learner is not benefiting from the additional ambiguous instances in the *Integrated* strategy.

We also found that an additional *HighAgree* training instance is generally more valuable that an additional *Integrated* training instance, especially with large training data sizes. However, in tasks where the *HighAgree* strategy may be causing uneven downsampling between classes, at lower training data sizes, *Integrated* instances may be more valuable, because they are members of underrepresented classes.

Beigman Klebanov and Beigman (2014) have previously found that, in a single task of classifying words in a text as semantically *new* or *old*, the inclusion of low item agreement instances in the training data caused a performance drop on high agreement test cases. The results of our investigation of five natural language tasks do not disagree with this finding;

however, their work investigated only a single task, limiting the generalizability of their finding. Our five natural language tasks were selected based on their diverse statistical experiment paradigms; while it is reasonable to assume that our findings will generalize to other natural language tasks, this remains to be proven in future work.

While similar investigations on training the best classifier from multiple labels may be relevant to other sources of annotation besides crowdsourcing (such as multiple expert annotations, games with a purpose, etc), different annotation sources foster different types of label noise, and our conclusions do not necessarily generalize beyond crowdsourcing. Additionally, our findings regarding the importance of a high number of item agreement levels for effective *HighAgree* use exclude most expert annotation tasks, which usually obtain less than five rounds of labels.

Although it is not possible to apply the training strategies from this chapter to the thread reconstruction corpus discussed in Chapter 3 (Ecd) because it is a pilot study and too small for thread disentanglement (only 100 email pairs), we found no evidence that the conclusions obtained from the five tasks in this chapter would not generalize to a thread disambiguation task.

In this chapter, we have compared several techniques of how to train a machine classifier on noisy crowdsourced labels. In the previous chapter (Chapter 3), we have contributed solutions towards problems faced in annotating an inevitably class-imbalanced discussion thread corpus. The current chapter, combined with Chapter 3, enables thread reconstruction research, as well as research on other natural language tasks with class-imbalanced or crowdsource-labeled corpora, by solving crowdsource-label problems with the annotation and machine learning stages of the task.

Part II

Experiments in Thread Reconstruction

Chapter 5

Thread Reconstruction

In this thesis, we undertake the task of content-based discussion thread reconstruction. This task is anchored in a variety of other thread research, from discussion post and user modeling to thread summarization. Yet discussion thread reconstruction is simultaneously unique to the wide variety of source online applications and their inherent problems.

In this chapter, we provide the theoretical foundations of discussion thread reconstruction. Section 5.1 will define concepts of thread reconstruction and provide examples of online discussion threads and related problems. In Section 5.2, we discuss previous research that is related to discussion threads and thread reconstruction. In Section 5.3, we explain the construction of our Enron Threads Corpus and the English Wikipedia Discussions Corpus (Ferschke, 2014). We will use the Enron Threads Corpus (Etc) in experiments described in Chapters 6 and 8, and we will use the English Wikipedia Discussions Corpus (Ewdc) in experiments described in Chapters 7 and 8.

5.1 Overview

A *discussion* is any conversation between two or more people. A *discussion thread* is the entire discussion viewed as a whole in written (usually online) form, after it has occurred; *thread* refers to the chaining together of the *discussion turns* (one speaker's contribution, such as one email or one forum post) to show the linear order in which they occurred. Although many discussion threads can be modeled as a single chain of adjacency (reply-to) relations, a discussion may sometimes split when one discussion turn receives multiple replies. Figures 5.1 and 5.2 show an email thread that can be modeled as a single chain: each discussion turn (here, an email) is a reply to the most recent email. Figures 5.3 and 5.4 show a Wikipedia discussion thread which contains multiple replies to the same turn, splitting the discussion thread into a branching tree.

Email1: *oh.....my.....god.....*
Email2: *what??????????*
Email3: *What's up?*
Email4: *I am in vacation mode. I don't want to do anything. what are you doing?*
Email5: *I've got a meeting at 3:30 (for an hour at most) then I'm leaving for the day! tomorrow, in about 10, out at noon. I can't wait.*

Figure 5.1: A discussion thread from the ETC that can be modeled as a single chain.

Figure 5.2: The discussion from Figure 5.1, represented as a graph.

Topic: "Grammatical Tense:gutted"
Turn1: *This article has been gutted. I deleted a lot of the cruft that had taken over, but a lot of former material is missing.[...]*
Turn2: *Good; the further this nest of doctrinaire obscurities is gutted, the better.*
Turn3: *Wait, you changed it to say that English doesn't have a future tense or you're citing that as an error (which it would naturally be)? For what it matters, [...]*
Turn4: *English doesn't have a future tense. It indicates the future with a modal (will) used with the present-tense inflection of the verb. [...]*

Figure 5.3: An EWDC discussion thread that must be modeled with a branching tree structure.

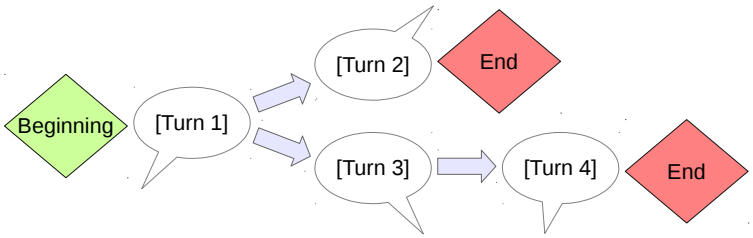

Figure 5.4: The discussion from Figure 5.3, represented as a graph.

To model branching discussion threads like the tree shown in Figures 5.3 and 5.4, in this thesis, we formally define a discussion thread as a directed, rooted graph (not chain) consisting of two or more discussion turns contributed by two or more participants, such that each participant contributes at least one discussion turn. In the graph, each node is a discussion turn, and each edge is a reply-to relation between pairs of turns.

5.1.1 Examples of Discussion threads

Discussion threads are found in a variety of applications online. In this section, we describe some types of discussion threads that are common in Web 2.0.

Emails Email threads consist of emails sent in reply to other emails, as well as emails forwarded to a new participant. There is frequently a small delay (hours or days) between sending and receiving an email, with the result that email messages tend to be longer than turns from *synchronous* (non-time-delayed) discussions, and provide more background context, so that the addressed participant can remember the topic of discussion. Email threads are viewed in an *email client*, computer software used to access and manage email. Some clients display emails as lists of threads, while others list individual emails in the order received. Clients use different rules to determine email thread structure: while many emails contain an `In-Reply-To` header, which identifies a email's parent email in the thread, thread structure may also be determined by matching `Subject` headers, similar `Date` headers, shared quoted material of previous emails, etc. An email sender specifically directs an email message at one or more intended receivers (i.e., *one-to-one interaction* or *one-to-few interaction*); therefore, a sender can ask direct questions and make impositions of the receiver that are not possible in threads of undirected turns, such as news website comments.

The ways in which email thread structure breaks down can be divided into three groups. *Real-time user errors* include mistakes such as: a user clicks on the wrong email but the right intended recipient, to send a reply; discussion participants continue to `Reply-to` emails from a discussion long after the topic of the discussion has shifted, with the unintended consequence of concatenating two or more threads; or, discussion participants continue to `Reply-to` emails from a discussion when discussing multiple different sub-topics of the original thread, with the unintended consequence that the reader cannot determine the subtopic by means of the `Subject` header or participant list. *Email client errors* include mistakes such as: the browser auto-combines multiple threads for display because they have the same `Subject` header; the browser fails to auto-cluster emails into the same thread because the `Subject` headers are different (i.e., the other participant's email client adds `AW:` to the `Subject` line of an email reply); or, the browser fails to save all email headers along with the email document, and later retrieval produces email documents without headers (e.g., the Enron Email Corpus). Finally, *user misuse* include intentional client usage that results in missing or wrong thread structure: email

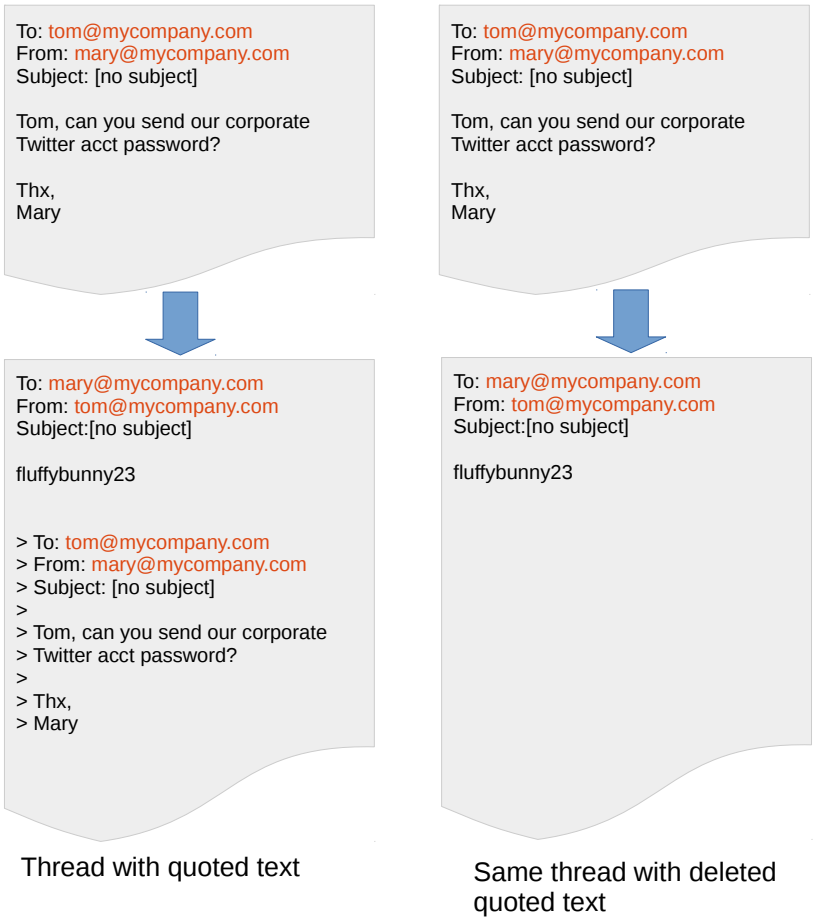

Figure 5.5: Example of a thread with and without quoted text. A user may delete quoted text to confuse the thread structure, making the thread more difficult for a third party to read.

communication via email drafts saved to a shared account (such as the 2012 case of U.S. General Petraeus); emails sent with fraudulent headers (a common tactic of email spammers); user deletion of quoted text from the previous email in the thread (see Figure 5.5); or, email account manipulation such as many accounts for one user or multiple users sharing one account.

Internet Relay Chat IRC chats consist of instant messages sent via software application to specific other users. Conversation participants can view the entire chat as a scrolling text window. Chat messages are usually sent and received instantly, so the text is generally short and provides no background context or additional world knowledge to help the user remember the topic of the chat. Chat messages are displayed in the order they are received; the software does not attempt to cluster messages by subject or thread except as explicitly determined by the chat participants. Because the user directs a message to a particular participant (i.e., *one-*

Chanel: *Felicia: google works :)*
Gale: *Arlie: you guys have never worked in a factory before have you*
Gale: *Arlie: there's some real unethical stuff that goes on*
Regine: *hands Chanel a trophy*
Arlie: *Gale, of course ... thats how they make money*
Gale: *and people lose limbs or get killed*
Felicia: *excellent*

Figure 5.6: Sample IRC chat (Elsner and Charniak, 2010, p. 390)

to-one or *few-to-few*), the user can ask questions or make requests that would not be possible in a more general chat forum.

As Elsner and Charniak (2010), explain in detail, the lack of auto-clustering messages by thread can make it difficult for a participant to understand which topic a message concerns. This is illustrated by Elsner and Charniak (2010)'s sample IRC chat in Figure 5.6. In this chat, it is unclear if Felicia's message ("*excellent*") is a reply to Gale ("*and people lose limbs or get killed*") or Chanel ("*Felicia: google works :)*").

Wikipedia discussion pages Each Wikipedia article has a forum for discussions about how the corresponding Wikipedia page should be written/edited/formatted, etc. A Wikipedia discussion page is open to public commentary, and users post comments by editing the Wiki markup of the discussion page. Because a discussion in this markup is a single file, a user can add, change, or delete content from any part of the discussion, with no mechanisms to ensure a turn is well-formed or added to the correct place.

Turns may be long or short. Some turns, like "*Done.*" are posted as a reply to a message requesting a change in the article. Other turns, such as turns arguing for a particular change in the article, might be multiple paragraphs. Due to the public commentary nature of the discussion pages (many-to-many interaction), it is necessary for a user to make their argument effectively and succinctly in a single turn, or they risk alienating other users with vague or misinterpreted messages.

There may or may not be a direct response to a user's post; such is the nature of the public forum. A participant may join the discussion and leave unpredictably, with responses posted minutes or weeks later. To avoid a participant's departure causing a discussion to be abandoned, user often post messages written as open-ended suggestions or requests aimed at no one in particular. This language style is very different from messages aimed at a specific participant. We discuss this stylistic difference and provide examples in Section 7.2.2.

Threads are displayed as a textbox of the entire discussion, with indents (notated by the character ":" in Wiki markup) indicating reply-to relations. Participants frequently mis-indent their turns. In Section 5.3.3 , we find that in an analysis of 5 random Wikipedia discussion

Turn1: *This article has been gutted. I deleted a lot of the cruft that had taken over, but a lot of former material is missing.[...]*

Turn2: *Good; the further this nest of doctrinaire obscurities is gutted, the better.*

Turn3: *Wait, you changed it to say that English doesn't have a future tense or you're citing that as an error (which it would naturally be)? For what it matters, [...]*

Turn4: *English doesn't have a future tense. It indicates the future with a modal (will) used with the present-tense inflection of the verb. [...]*

Figure 5.7: The discussion thread from Figure 5.3, displayed with its original, wrong thread structure.

threads longer than 10 turns each, 29 of 74 total turns, or 39%±14pp of an average thread, had indentation that misidentified the turn to which they were a reply. Turns whose tabbing correctly identified a parent turn but whose parent misidentified an ancestor turn were counted as correct for this analysis. We also found that the misindentation existed in both directions: an approximately equal number of tabs and tab deletions were needed in each article to correct the mis-indented turns. The original, wrong indentation of the sample Wikipedia discussion from Figure 5.3 is shown in Figure 5.7. In this discussion, Turn 3 is clearly a reply to Turn 1, but it is marked as a brand new message that is not a reply to any turn.

Social voting sites Social voting sites[61], such as Reddit (Weninger et al., 2013) or Slashdot, provide a different sort of discussion style. The structure of these websites is an online bulletin board, where users submit content or direct links, and other users can submit comments about the topic. A discussion consists of comments (discussion turns), organized in reply-to relations (denoted by text indent), with the entire discussion visible on the website. A screenshot of a Reddit discussion is shown in Figure 5.8. Both Reddit and Slashdot offer a variety of comment screening and sorting tools.

A key feature is that users vote up or down both the posted items and also the comments, and the discussion display changes in real time to reflect the re-ordering from the votes. Bad or unpopular comments are downvoted until they are hidden, and popular comments are upvoted until they are displayed at the top of the discussion page. Discussions can have extensive tree structures, and the hierarchy is preserved while the order of branches is rearranged in response to voting. Users of Reddit, called *Redditors*, are credited with points as a reward for receiving upvotes on their comments. This motivates Redditors to contribute the best quality comments that are likely to be popular. Comment length ranges from very short (a single word or less) to long (multiple paragraphs). Comments are visible instantaneously. Although comments are posted publicly like Wikipedia discussion turns (in a many-to-many interaction), Redditors

[61]Terminology from Gilbert (2013)

seem to frequently respond to comments aimed at them; responses to responses may serve to increase the vote score of their original comment, increasing the Redditor's total point score.

Changing the display of a discussion tree structure based on the voted popularity of branches has the unusual effect of changing the popular discussion thread, as seen by later participants; a form of "discussion editing". With this technique, errors that exist in other online discussions, such as comments submitted to replies of the wrong previous comment, are quickly identified and downvoted to invisibility in popular discussions. However, it would be helpful if this behavior were available via automated tool, so that it could also be applied to unpopular threads, which don't get enough viewers to be effectively edited.

Another source of user error is the voting procedure; the same discourse model of thread structure that can identify incorrect reply-to relations could also be used in conjunction with votes to identify, in real time, that a user just clicked a vote different from what they had intended, or had accidentally voted when they intended to scroll the screen, etc.

While Redditor scores do not have much use in the real world (i.e., outside of Reddit), Redditors take them very seriously, and some Redditors manage multiple accounts to manipulate their scores (Alfonso III, 2014). For example, a Redditor with multiple accounts may use one account to post content, and another account to upvote that content. Other Redditors are more likely to view a post that already has upvotes, and a post that has more views is more likely to receive more additional upvotes. This may have real-world impact when Redditors decide to send money or other forms of help to strangers, based on the content of posts (Gulino, 2015).

News articles The comment sections of news articles are a form of discussion in which readers of news articles online (New York Times, CNN, ABC, Al Jazeera, Time, and others) can post comments at the bottom of the webpage after reading a news article. It is usually also possible to respond to others comments, creating a tree structure discussion thread, although the majority of news website posts are in direct response to the article. Turns may be long or short, but some news websites reject submitted posts over a certain length in a bid to improve discussion quality. Posts are usually also screened for offensive language and ideology. Posts are displayed near-instantaneously, but there may be a delay of a couple minutes while the post is screened. Users often have the option of voting posts up or down, but many news websites still have lots of spam and troll posts, and when posts are displayed in the order received, it is unclear how voting changes the display. (The algorithm used in the display is proprietary.) Most posts are not aimed at specific users, but when a post is directed at one specific user, it is quite rare for that user to respond, perhaps due to the widespread low quality (spam and troll posts, and little or no voting tree restructuring) of the discussions. In general, news article posts are a forum for users to vent their feelings on a political issue, or current event, or racial/religious/social group, without productive discussion beyond the emotional venting. A screenshot of the comments section from a news article website is shown in Figure 5.9.

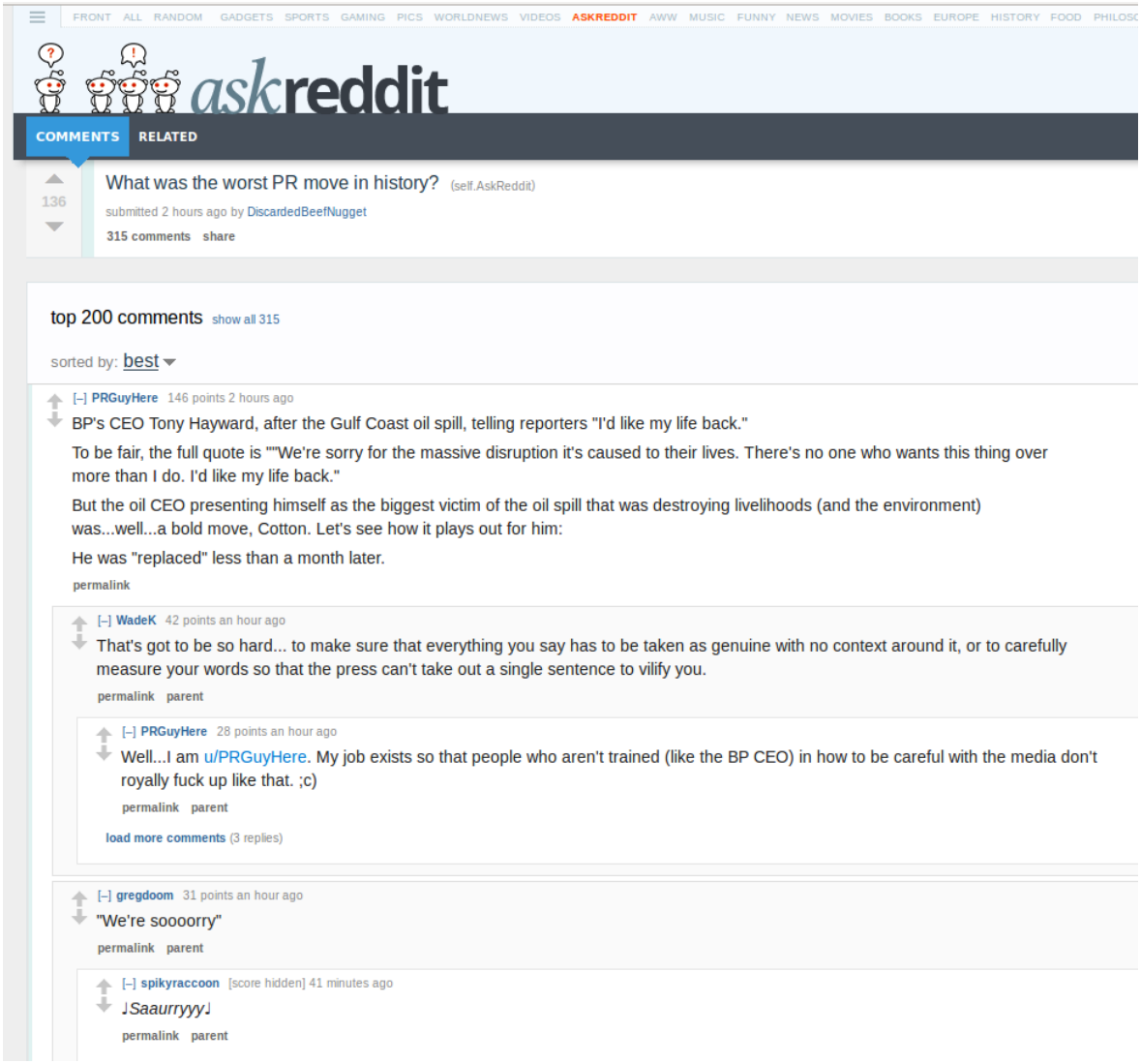

Figure 5.8: A screenshot of a discussion in the voted forum Reddit.

Like Wikipedia discussion forums, users of news website forums frequently reply to the wrong comment. In Wikipedia discussion forums, reply-to relations are signaled in the Wiki markup language via a character (":") indent. On news article websites, reply-to relations are created when the user clicks a `Reply` button for a particular comment. When the user clicks the button for the wrong comment, the wrong thread structure is created.

Question-answering websites These sites contain discussions that result from an original question posted by a user. While format varies from website to website, a universal feature is a series of ranked answers posted in reply to the user's question. Users can submit new answers, and answers are up- or down-voted. Answers are displayed as a list below the question, with visible ratings, in order of popularity. There is no mechanism for the question-submitter to address a particular answerer; anyone from the community contributes answers (one-to-many interaction).

Question-answering websites can benefit from discussion thread structure to assist the voting procedure in ranking the answers. There is randomness in relying on the votes of readers, and an automatic system that can recognize good and bad answers for a question can re-order the answers to display a better order. Consider the screenshot in Figure 5.10. In this example, none of the answers have received any votes yet. It is necessary to use some other discussion model to decide which order to display the answers, until the answers receive votes. Additionally, when the answers have only received one or two votes, it is possible that these votes will be noisy and will not effectively select the best answer (i.e., perhaps one of the voters clicked a wrong answer by accident, or perhaps one of the voters did not understand which answer was correct, etc.).

In this section, we have shown that online discussion threads are present in a range of applications and uses. We have listed a number of examples, including emails, IRC chats, Wikipedia discussion pages, social voting websites, news article comments, and question-answering websites. We have described components of these discussions that rely on thread structure, and where thread structure errors may occur or where the application may otherwise benefit from thread structure modeling.

In this thesis, we will work to reconstruct the threads of two of these types of discussions: emails and Wikipedia discussion pages. We have chosen to investigate these types of discussions because the available corpora permitted strongest investigation for each relevant subtask of thread reconstruction: the wide domain coverage of the Enron Emails Corpus (the email source for out Etc), with its many topics from football to trips to meetings, is a challenging test case for thread disentanglement; and the Ewdc, which consists of direct replies (the Etc contains other relations as well, such as *forwarding*) is a well-formed test case for adjacency recognition.

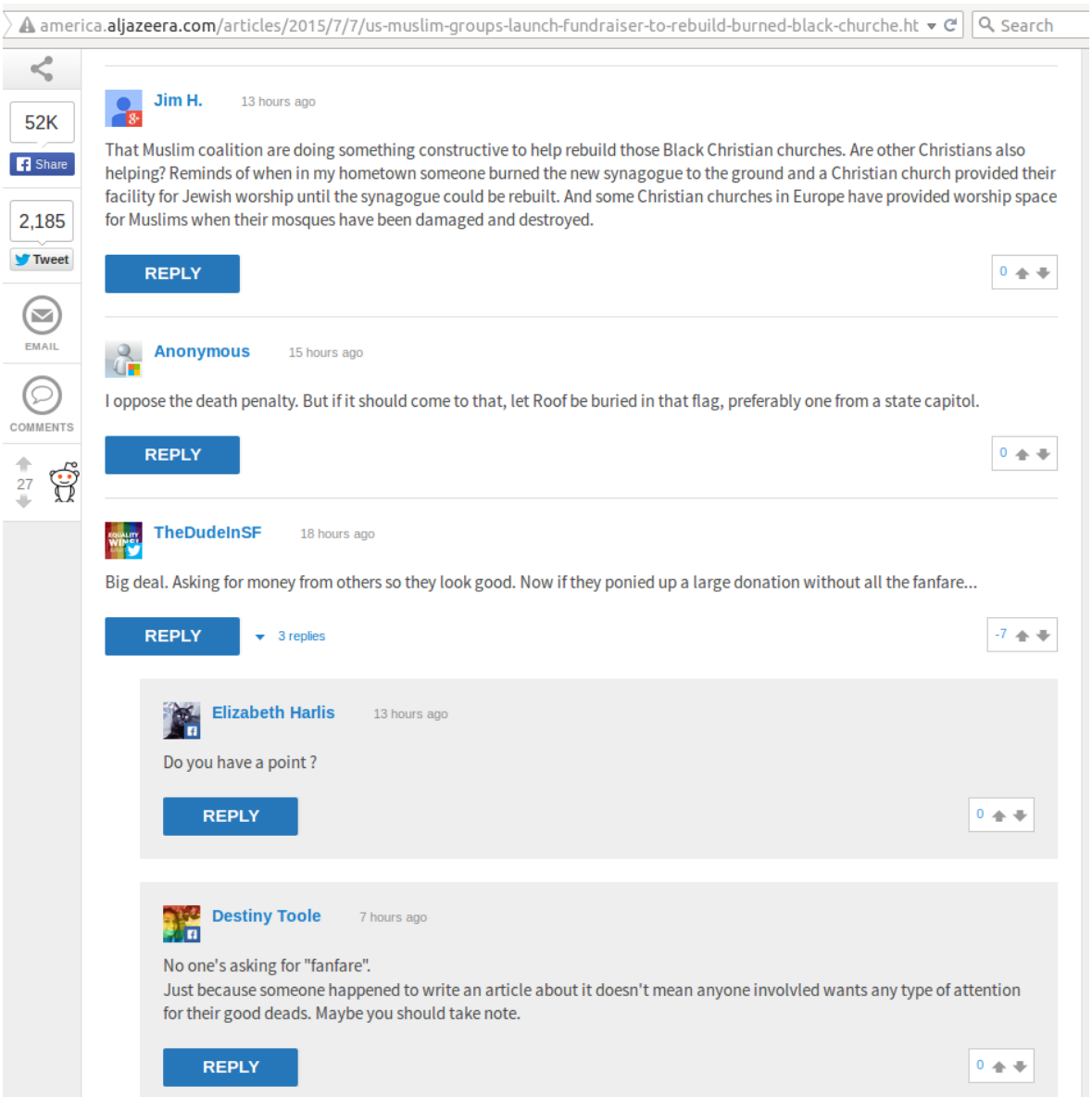

Figure 5.9: A screenshot of the comments section of a news article webpage (Al Jazeera).

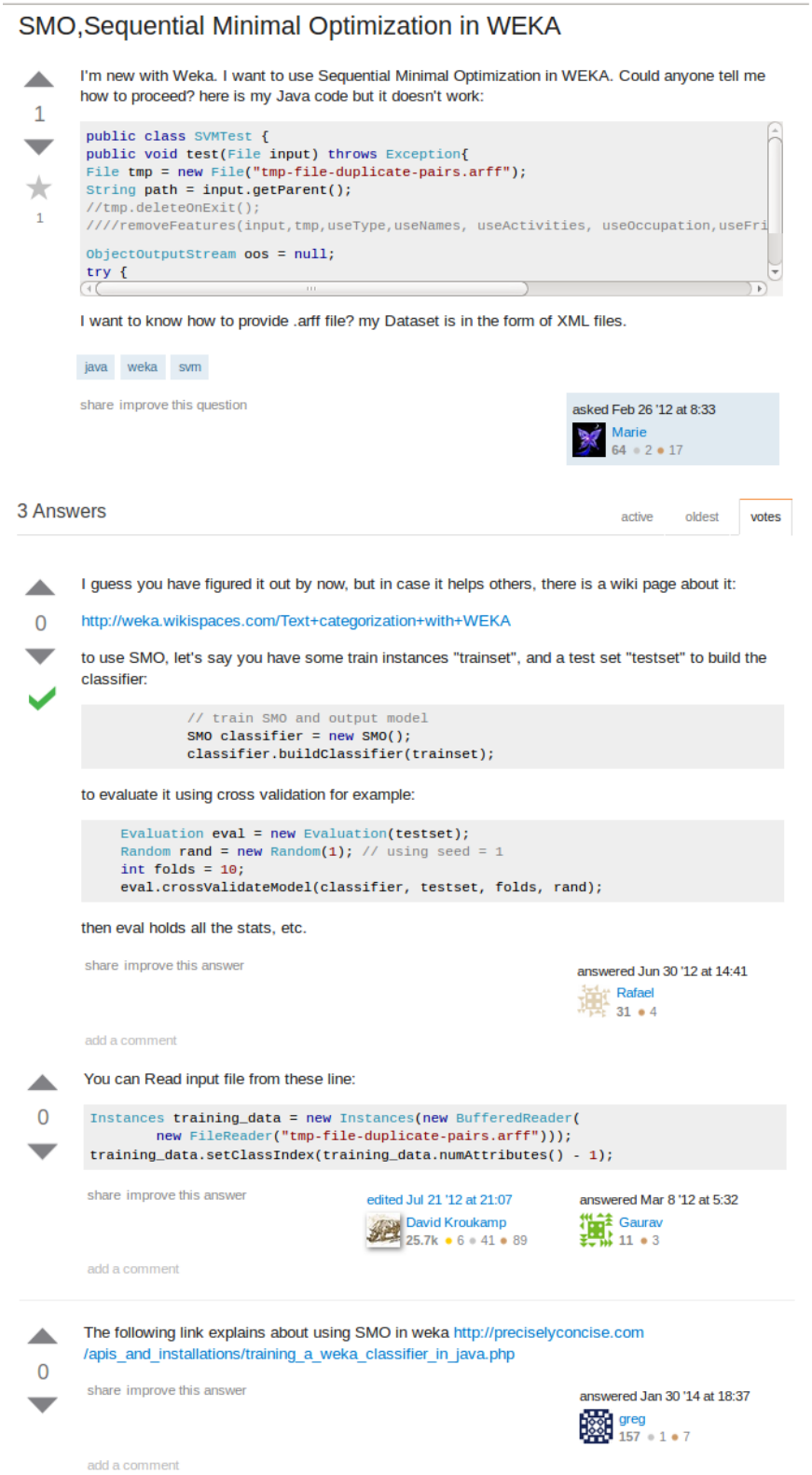

Figure 5.10: A screenshot of a question-answering webpage from Stack Overflow. None of the answers has received any votes, so another algorithm must be used to determine display order.

Figure 5.11: A comparison of thread disentanglement datasets with few threads versus many threads. The dataset with many threads has a much higher negative class prior, as shown by the larger number of red edges.

Figure 5.12: A comparison of adjacency recognition datasets with few emails in a thread versus many emails in a thread. The dataset with many emails in a thread has a much higher negative class prior, as shown by the larger number of red edges.

5.1.2 Thread Reconstruction as a Task

The overall research goal of this thesis is discussion thread reconstruction. In this task, we determine the discussion thread relations between an unordered, unsorted bag of discussion turns.

While other NLP tasks such as POS tagging and sentiment analysis maintain relatively stable class priors for different corpora within the same domain, the class priors of thread reconstruction datasets change wildly. If our bag of discussion turns contains only turns from two discussions, then reconstruction is easier than if the bag of discussion turns contains turns from 1,000 conversations. If our bag of discussion turns contains only turns from a conversation of 2 turns, then the reconstruction is easier than if the bag of discussion turns contains turns from a conversation of 45 turns. These two scenarios are illustrated in Figure 5.11 and Figure 5.12 respectively. In both scenarios, the change of class priors can be seen by the higher percentage of red (negative) edges between email nodes.

In Section 5.1.1, we described a number of applications and uses in which discussion threads are present, and in which thread structure may be lost, or additional structure may be desirable. In many of these potential use cases, thread reconstruction faces discussion turns from a large number of threads and/or a high number of turns. Therefore, we proceed by planning for the worse case scenarios: we anticipate an unordered, unsorted bag of discussion turns with turns from multiple conversations with potentially many turns each, and all with no available metadata.

In this thesis, we break down the task of discussion thread reconstruction, into 3 subtasks: (1) discussion thread disentanglement, (2) adjacency recognition, and (3) thread graph construction. Discussion thread disentanglement is the task of separating our bag of turns into sets of turns for each discussion. Adjacency recognition is the task of identifying reply-to relations between turns of the same discussion. Thread graph construction uses the adjacency relations determined in adjacency recognition to build (and prune) the full discussion graph. Discussion thread disentanglement and adjacency recognition are natural language processing tasks; thread graph construction is a computer science graph optimization problem, and uses no NLP. Thread graph construction lies outside our research domain, and we leave this task to other researchers.

5.1.3 Alternatives to Disentanglement/Adjacency

There are a number of relations that could be modeled during thread reconstruction, as alternatives to thread disentanglement and adjacency recognition. Besides reply-to relations, some turns reply to one turn while acknowledging another. This is shown in Figure 5.13, where *Turn12* acknowledges the list of contributions in *Turns 3–11*.

At other times two turns contribute towards the discussion as a single turn. In Figure 5.14, *User4*'s turn functions as an extension of *User3*'s turn.

Sometimes multiple people will contribute to a discussion as though they are the same person. This is shown in Figure 5.15, where *User4* replies as though they are *User2*.

Additionally, a discussion may switch to discussing the discussion itself. While not a separate thread, clearly a topic shift has occurred. An example is shown in Figure 5.16, when *User4* makes a joke about the discussion itself.

While such relations could also be used to reconstruct discussion threads, and undoubtedly will be the topic of insightful future investigations, to the best of our knowledge there is currently no work in these areas. We followed the precedent of previous work in choosing the subtopics of thread disentanglement and adjacency recognition for the research in this thesis.

5.1.4 Pairwise Evaluation

We chose to use pairwise evaluation for measuring system performance. Such evaluations can be measured with common metrics such as *F-measure* and *Accuracy*. While pairwise

Turn1: *? It's certainly the most intense U.S. Pacific hurricane. These should be mentioned (probably in the intro and storm history). Should we have tables for these things (like tl|Costliest US Atlantic hurricanes)?*

Turn2: *Officially, it is the costliest, but some hurricanes, like Pauline, might have caused more (their USD amounts are unknown). These are the costliest EPAC tropical cyclones with a damage total of over $1 million (2005 USD).*

Turn3: *Hurricane Iniki- $3 billion (Hawaii)*
**Hurricane Iwa- $507 million (Hawaii)*

Turn4: *Hurricane Kathleen (1976)- $137-$549 million (California)- I just used the average for $543*

Turn5: *Hurricane Norma (1981)- $300 million*
[...]

Turn12: *So I guess it's the costliest U.S. Pacific hurricane? And, what's the source for this list?*

Figure 5.13: Wikipedia discussion costliest.mostintense from Discussion:Hurricane$_I$*niki*

User1: *What's a skill you think everyone should have but most people don't?*

User2: *Cook at an average level. Frozen pizza, grilled cheese (or melts), and pasta does not count as cooking.*

User3: *It counts for me!*

User4: *As long as we can keep ourselves alive!*

Figure 5.14: Reddit discussion[62] in which *User4*'s turn is an extension of *User3*'s turn.

User1: *What was totally NOT A PHASE for you, but ended up being a phase?*

User2: *Angsty Poetry. Great if you're 15. Not so much when you're no longer a teenager.*

User3: *Can you give us an example?*

User4: *why does the world*
believe in god
but not in me
and the girl
who sits beside
me in class
together
god is my friendzone

User2: *Nailed it.*

Figure 5.15: Reddit discussion[63] in which *User4* replies as though they are User2.

User1: *What's a skill you think everyone should have but most people don't?*
User2: *TURN SIGNALS TURN SIGNALS TURN SIGNALS*
User3: *Or the guy that has the turn signal on for 2 hours straight.*
User4: *That's the guy above you.*

Figure 5.16: Reddit discussion[64] in which *User4*'s turn is an extension of *User3*'s turn.

evaluation introduces the evaluation artifact that one turn can be classified as a member of two different other threads, it avoids the controversy of cluster evaluation metrics. As Amigó et al. (2009), in an evaluation comparing different cluster metrics, point out, "determining the distance between both clustering solution (the system output and the gold standard) is non-trivial and still subject to discussion." Different cluster evaluation metrics favor different types of clusters. Strehl (2002) finds that the metrics *purity* and *entropy* are biased towards smaller clusters, and *F-measure* (of clusters) is biased towards coarser clusters. Strehl (2002) recommends *Mutual Information* as unbiased and symmetric, but Amigó et al. (2009) and Meila (2003) point out other constraints that a metric should satisfy, and Amigó et al. (2009) find *BCubed precision* and *recall* to be the only major metrics which satisfy these constraints. By our use of pairwise evaluation, such controversy is minimized.

5.2 Background

In this section, we provide a foundation of previous research on discussion threads. In Section 5.2.1, we consider previous research on discussions. In Section 5.2.2, we give a background on discussion thread research. In Section 5.2.3, we specifically discuss thread reconstruction research.

5.2.1 Discussion: Related Work

The historical study of *discussion* (or conversation) is extensive and involves the research communities of discourse analysis in linguistics, conversational analysis in sociology, and conversational psychology in psychology, among others. We will provide only the briefest overview of the field.

A conversation consists of at least two people talking together. There is no definition that is widely accepted that is more specific (Warren, 2006). Svennevig (1999) references the necessity of discussion turns with his definition:

> "Conversation is a joint activity consisting of participatory actions predominately in the form of spoken utterances produced successively and extemporaneously by

> different participants in alternating turns at talk which are managed and sequentially organized."

In their foundational paper on the organization of turn-taking, Sacks et al. (1974) state that the organization of taking turns to talk is fundamental to conversation and other forms of speech exchange. Sacks et al. propose a model in which the organization of turn-taking is locally-managed, party-administered, interactionally controlled, and sensitive to recipient design. In other words, turn-taking organization is not determined or determinable at a different time, nor by non-participants, and it must dynamically respond to new turns and information and the desires of the participants. An example type of activity of at least two people talking together which fails these requirements is a scripted theatrical event. And following Sacks et al. turn-taking model, Schegloff and Sacks (1973) point out that a conversational unit does not simply end (as a theatrical script can end) but is brought to a close.

The work in this thesis follows the theoretical foundations that a discussion must be "locally-managed, party-administered, interactionally controlled, and sensitive to recipient design", and presumes that a discussion reflects the local and dynamically changing environment of the participants. Our thread disentanglement investigation in Chapter 6 makes no presumptions of conversational organization beyond token similarity and (psychologically motivated) speaker accommodation. Our adjacency recognition follows directly from the work of Sacks et al. (1974) and others, in presuming that the fundamental conversational unit within the discussion is the adjacency pair, where the speaker (or writer, etc.) is dynamically reacting to the turn of the previous speaker.

5.2.2 Threads: Related Work

Software Previous work has examined a number of properties of discussion threads. Different software tools have been proposed, providing various discussion thread display visualizations.

Smith and Fiore (2001) describe new software, the Netscan dashboard, developed for visualizing a persistent messaging system. Examples and results of user studies are reported for use with Usenet. The visualization components in this software include a visualized thread tree, a *piano roll* to show participation amount of various users in the current thread, a sociogram showing interpersonal connections, as well as a standard message display pane to show the current selected message. Users found the thread tree interesting but of conflicting usefulness; the piano roll was merely confusing, and the sociogram was interesting but confusing to use (Smith and Fiore, 2001).

Guzdial and Turns (2000) present CaMILE, a discussion forum tool for computer-mediated anchored discussion. Compared with a baseline newsgroup forum, CaMILE discussions had significantly longer thread length. The paper also found that anchored discussions, or discus-

sions "beginning with a document or topic which students may be interested in discussing," scored higher in outcome measures of discussion effectiveness than unanchored discussions.

Giguet and Lucas (2009) present Anagora, a tool to display discussion threads as a function of time. The tool is intended for use by instructors to monitor educational forums. A user study showed that, although this tool is less sophisticated than others available for forum monitoring, it is fairly easy to use.

Venolia and Neustaedter (2003) present a mixed-model visualization tool that simultaneously displays an email thread as a chronological sequence of messages and as a tree-structure representing reply-to relations. Results from a user study show six users found both the chronological sequence and the tree-structure easy to read.

Viégas et al. (2006) present *Themail*, an email archive visualization tool that displays keywords (identified by TF-IDF weighting) of an email user's correspondence with a particular other individual in a list format as a function of time. This shows how the topic of discussion changed over time. Results of a user study showed that some users prefer analyzing the big picture of topic changes, while others prefer identifying particular conversations via keywords.

Kerr (2003) presents an email thread visualization technique to assist users in locating a particular email within a thread. *Thread Arcs* displays visual arcs as reply-to relations between temporally-ordered nodes representing emails. Users liked the result, which was more compact than thread trees, and seven of the eight users said they would like a thread visualization in their future email clients.

Diep and Jacob (2004) present an email interface system with unique properties for visualizing email messages and threads. While a traditional email client displays emails or threads in a date-sorted list of sender/subject metadata, this paper's system displays each email as rectangular objects that may be dragged around the screen space. There is also a zoom option with semantic features, where different zoom levels display different amounts of keywords in each rectangular email object.

Doran et al. (2012) present a tool, "CoFi", for viewing and analyzing forum discussions threads, which are typically not indexed by search engines. This tool assists the user by harvesting discussion comments from a website, automatically clustering them by topic, and producing a visual display of analysis such as most popular words, most active users, etc.

Modeling the discussion Previous work has modeled or predicted characteristics of the discussion itself.

Gómez et al. (2008) examine the thread structure of discussions on `slashdot.org`. Results show that the discussions show heterogeneity and self-similarity throughout the different levels of the discussion.

Jeong (2005) examines the effect of message labels on response rate and response time in educational forum discussions. It was found that *argument-critique* message pairs gener-

ated 2.88 more subsequent replies, and the probability of a response was higher than for *critique* messages alone. *Evaluative* message types received the fastest responses. *Critiques* had a significantly longer wait time than other message types, but were most likely to generate a response.

Hewitt (2003) examines the impact of topic ("note") date on response rate in an educational forum. It was found that students tend to reply to the most recently introduced topics ("conferences") from the instructor, and ignore the older topics. The author notes that this may have a detrimental educational impact of drawing attention away from important issues being discussed in older threads.

Gómez et al. (2013) present a model to analyze the structure and evolution of discussion threads. On discussions from four different popular websites, a parametric generative model uses popularity, novelty, and trend/bias features to reply to the thread originator. The experiments show that the model with these features is able to capture many statistical properties of the discussion threads.

Hewitt (2005) investigates the causes of thread death (e.g., abandonment of the discussion) in an online classroom forum. It was found that about half of thread death is attributed to the instructor posting a new "conference" (topic for discussion), causing old threads to be abandoned. They also found that thread death is associated with the usage practice of only reading new messages when choosing messages to respond to; even in an artificial model where all messages have equal content, it was found that this practice causes most threads to die quickly.

Thread summarization One subtopic of thread discussion research is email thread summarization. Email thread summarization mediates the time-consuming task of responding to a backlog of email, by providing the end results of a discussion that may have taken many turns to negotiate.

Wan and McKeown (2004) propose an email thread summarization system specifically aimed at discussions of a group to arrive at a consensus agreement. The sentence-extraction-based system uses a language model over the reply emails to rank sentences in the initial email, with the highest-ranking sentence extracted as a summary of the thread *issue*. The first sentence of each reply email is extracted to summarize the responses. An evaluation showed that the methods proposed outperform baselines of simply extracting the first n (where $1 \leqslant n \leqslant 3$) sentences of the initial email.

Rambow et al. (2004) present an algorithm for summarizing threads via sentence extraction. A gold standard was constructed by having annotators write a human summary of each thread, and then each sentence in the thread was compared with the human-written summary via sentence similarity; high-scoring sentences were gold standard positive, and low-scoring sentences were gold standard negative. Then, a variety of similarity-based and thread-descriptive features are used to predict each sentence in the thread as positive or negative. An evaluation

showed that the thread specific and email-specific features improve system performance over features that treat the thread as one long single text.

Carenini et al. (2007) propose a framework for summarizing email conversations, called ClueWordSummarizer (CWS). In this framework, a fragment quotation graph is constructed to represent the reply-to relations between text fragments in the thread, including replies interspersed with quotations in a single email. *Clue Words*, or words that occur in two directly connected nodes in the graph, are used to weight the sentences for inclusion in the summary. Evaluation results show that CWS outperforms two other email summarization systems, MEAD and RIPPER.

Murray and Carenini (2008) extend the work in Carenini et al. (2007) by constructing their fragment quotation graph. The proposed system extracts a variety of length-, structural-, and participant-based features of the fragments for a logistical regression classifier. An evaluation showed that this set of features is effective for both recorded meeting transcripts and email threads.

Carenini et al. (2008), working on email conversation summarization, extend the concept of a fragment quotation graph (Carenini et al., 2007) to the sentence level, investigating graph edge weighting via Clue Words (Carenini et al., 2007), WordNet semantic similarity, or cosine similarity. Sentences are selected for inclusion in the thread summary based on their high score, as calculated by the ClueWordSummarizer (CWS) or PageRank algorithms. Evaluation results showed that CWS outperforms several forms of PageRank.

Modeling the user Some previous work characterizes properties of users, the discussion participants.

Dasigi et al. (2012) investigate subgroup detection in online discussion threads. Subgroups are clusters of users that share a similar opinion on the debate topic at hand. In genre-independent studies, this paper found that opinion-mining is particularly effective in informed threads, such as discussions in `createdebate.com`. Implicit attitude features (LDA-based topic modeling of the sparse n-grams in short texts) are particularly effective in formal-style threads, such as Wikipedia discussions.

Welser et al. (2007) provide an analysis into the visual graphical signature of one type of participant in online discussion groups, namely, *the answer person*. They find that answer people tend to post replies to other isolated users (who also reply to few other authors); in their graphical networks, their neighbors are not neighbors of each other (e.g., a low proportion of three-cycles) ; and finally, they tend to reply to threads started by others, and often contribute only one or two messages per thread.

Daniil et al. (2012) propose an algorithm to issue a score for users on a forum, based on their number of posts, the weighted average score of their posts, the weighted average score of the threads that they posted in, and their social involvement (helpful replies, etc). Such a scoring

system is needed in educational forums, to help users learn to become better participants and to help facilitators find and reward the most important users, among other uses.

Bergstrom (2011) is a case study on trolling and the Reddit user "Grandpa Wiggly"/"Word-Sauce". A troll is a widely-acknowledged label for a forum user with "less than benevolent intents." This paper discusses the case of a Reddit user who misrepresented his identity but apparently didn't intend to cause harm, and illustrates how the label "troll" has become applicable to any user whose behavior runs afoul of community expectations, to the detriment to the forum community.

Modeling the post Other previous work characterizes properties of the individual posts, the discussion turns.

In Gómez et al. (2008)'s analysis of discussions on `slashdot.org`, classification is used to predict the degree of controversy of a post within a discussion. A simple measure of post controversy, based on the h-index measure commonly used to calculate the impact factor of a scientific paper, was found to be more predictive than a previous approach using structural and semantic information

Lin et al. (2009a) models message labeling in an educational forum as a cascaded text classification task, with the goal of providing better teacher support in moderating an educational forum. The results showed that the classifier cascade system significantly outperformed a non-cascade, decision-tree system.

Discussion constraints Previous work examines the effects of design-, automatic-, or moderator-imposed constraints on online discussions.

Brooks and Jeong (2006) investigate whether or not thread pre-structuring can improve the quality of discourse in educational forums, as evaluated on message-type-labeled comments. When students were required to post supporting and opposing arguments to separate designated threads, posts were found to elicit 64% more *challenges* than in unconstrained threads, although no changes were found in the numbers of *counter-challenges*, *supporting evidence*, and *explanations*. The paper proposes that these results support the use of thread pre-structuring to improve the quality of discourse.

Mazzolini and Maddison (2007) examine the impact of instructor participation on thread structure in educational forums. It was found that instructor initiation of a thread was correlated with shorter thread length and lower student posting rate, and instructor participation in a forum was associated with shorter average thread length, and lower student posting rate. An analysis of student course reviews found a correlation between instructor posting rate and perceived enthusiasm and expertise of the instructor, but not usefulness of the forum or overall satisfaction with the educational experience. They also found that the timing (posting throughout the duration or posting only at the end) of instructor posting did not influence student posting rates.

Gilbert and Dabbagh (2005) present a case study on a collection of education forum discussions, investigating the impact of the protocols and criteria that are used as community participation instructions to guide online discussions and create more meaningful discourse. Evaluation results indicate, among other findings, that greater facilitator participation led to increased participation overall and in-between students. However, the presence of posting protocols may have negatively impacted the cognitive quality of student postings, as evaluated by a quality system proposed in the paper.

Social voting sites like Reddit request users to vote content up or down, so that the best content is displayed most prominently. Gilbert (2013) investigates voter participation on Reddit, with results suggesting that not only is non-voting widespread, but significantly affects content ratings and site quality. It was found that 52% of the most popular links that were also reposts had been ignored during their initial post. These findings have implications on the thread structure of discussions on social voting sites, where the thread structure is displayed based on comment popularity: such threads may display lower quality discussion turns due to undervoting.

Downstream applications While many online discussions serve a purpose of entertainment alone, other discussions may impact a downstream purpose, such as educational quality and information retrieval.

Thomas (2002) presents an analysis of a study where undergraduate students participated in an educational discussion forum for a course. The paper argues that the forum was an ineffective learning environment, because "normal discussion" did not occur: many submitted messages were never read; even more messages received no response; and finally, the branching nature of discussion threads created "incoherent development of ideas."

Seo et al. (2009) investigate algorithms for information retrieval (IR) over forum threads. Because such thread structure is often inaccurate or missing, they implement a thread reconstruction technique using metadata and direct quotes. Experiment results show that when forum thread structure is present and accurate, thread structure use leads to improved IR.

5.2.3 Thread Reconstruction: Related Work

In this section, we provide an overview of work on various subtasks of thread reconstruction. This work is the closest related work to the overall goals of thread reconstruction of this thesis.

Elsner and Charniak (2010) investigate thread disentanglement of internet relay chat (IRC), a multi-party synchronous discussion platform in which users participate in multiple discussions in the same screen. A graph-based clustering model is used for chat disentanglement, based on lexical, timing, and discourse-based features. In the first stage of the model, a supervised linear classifier predicts whether or not pairs of turns are from the same discussion.

The resulting graph is partitioned in the second stage. The model consistently outperforms the highest-performing baselines.

Erera and Carmel (2008) perform conversation detection in emails (a form of corrected thread reconstruction) via a cluster-forming metric over message subjects, participants, date of submission, and message content. Experiment results show that all email attributes contribute towards better conversation detection, and that similarity clustering is effective for this task.

Yeh and Harnly (2006) present two techniques for email thread reassembly (a.k.a. reconstruction), as well as a technique for recovering missing messages that are suspected to exist in a thread. The first reassembly technique decodes the previously undocumented Microsoft Exchange header `thread-index`. The second reassembly technique uses string similarity over quoted sections of email, for the situation in which email headers are not available. Missing message recovery is based on predicted thread structure and quoted sections of email. In an experiment using the first technique as a gold standard, the second technique missed 7.3% of messages in threads; missing message recovery reduced this to 3.1%, showing that quotations are very helpful for thread reconstruction, when they are available.

Shrestha and McKeown (2004) investigate the detection of questions and answers in email threads, for the downstream purpose of better email thread summarization. Two approaches are proposed: first, for the automatic detection of questions in emails, and second, for the identification of answers from candidate sentence segments in subsequent emails. Experiment results show that the addition of thread specific features, such as the number of emails in between the question and candidate answer, produces better results than text similarity alone.

Joty et al. (2010) perform topic segmentation at the sentence level on email threads. Topic segmentation is necessary for higher-level analysis and applications including summarization, information extraction, and information retrieval. Experiments in this paper show that pre-existing LDA and LCSeg models are adaptable for conversational structure, and LCSeg is more effective than LDA for topic segmentation.

Aoki et al. (2003) investigate a mobile audio space system to assist participants in an audio discussion. In a text-based application supporting multiple discussions in the same conversational floor, text font characteristics could (theoretically) be used to signal a message's membership in a particular discussion. This paper proposes, for an equivalent audio system, that audio delivered to each participant should be modified "to enhance the salience" of turns from participants in the same current discussion. A user study found that participants appreciated this discussion-highlighting audio system, as long as it worked properly.

Several works use metadata to reconstruct forum threads. Aumayr et al. (2011) propose an algorithm to reconstruct discussion thread structure in online discussion boards that do not provide it. In an evaluation on the Irish forum `Boards.ie`, an algorithm using simple metadata-based features (reply distance, time difference, quotes, thread length) and cosine similarity with a decision-tree classifier is shown to accurately recreate a branching tree structure, and to significantly improve thread structure over a baseline algorithm. The experiments use a

similar stepwise process as we propose for thread reconstruction in this thesis: the algorithm first learns a pairwise classification model over a class-balanced set of turn pairs, and then uses the predicted classifications to construct graphs of the thread structures of discussions. Balali et al. (2014) use a text similarity-based feature, as well as a variety of metadata-based features, to learn a pairwise ranking classifier, and then construct graphs of the thread structures of news forum discussions. Wang et al. (2011a) also reconstruct forum discussion thread graphs using cosine similarity plus non-turn-content-based features, using discussions from Apple Discussion, Google Earth, and CNET.

Wang et al. (2008) reconstruct discussion threads from between players of the multi-player educational legislative game *LegSim*. They presume the metadata is unavailable, and use vector space model similarity between pairs of turns to identify *reply-to* relations to build the graphs. Three different proposed approaches modeling message temporal relationships in different ways all outperform a threshold-cutoff baseline.

5.3 Datasets

In our work on thread reconstruction, we use two datasets: our Enron Threads Corpus (Etc) and the English Wikipedia Discussions Corpus (Ewdc). In Section 5.3.1, we introduce and describe the Enron Threads Corpus. We provide examples, and discuss some of the issues faced during experiments using the Etc. In Section 5.3.3, we describe the English Wikipedia Discussions Corpus, and illustrate how we used it for thread reconstruction experiments. We provide examples, and discuss some of the issues faced during experiments using the Ewdc.

The Etc is a contribution of this thesis. Most of the material in Section 5.3.1 describing the creation of the Etc was previously published in peer-reviewed proceedings:

> **Emily K. Jamison** and Iryna Gurevych: 'Headerless, Quoteless, but not Hopeless? Using Pairwise Email Classification to Disentangle Email Threads', in: *Proceedings of 9th Conference on Recent Advances in Natural Language Processing* (RANLP 2013), Hissar, Bulgaria, 2013.

The Ewdc was created by (Ferschke, 2014). Most of the material in Section 5.3.3 describing our Ewdc dataset was previously published in peer-reviewed proceedings:

> **Emily K. Jamison** and Iryna Gurevych: 'Adjacency Pair Recognition in Wikipedia Discussions using Lexical Pairs', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing* (PACLIC), Phuket, Thailand, 2014.

5.3.1 Enron Threads Corpus

The Enron Email Corpus (EEC)[65] consists of the 517,424 emails (some of which are duplicates) that existed on the Enron Corporation's email server (i.e., other emails had been previously deleted, etc) when it was made public by the Federal Energy Regulatory Commission during its investigation of Enron. There are a wide variety of email types in the EEC: business emails, party invitations, news items, technical discussions, auto-generated logs, job applications, conference announcements, speaker invitations, football commentary, and personal emails both serious and chatty. There are 159 users' accounts, and 19,675 total senders (including non-Enron email senders).

The other two publicly available email corpora, the W3C Corpus[66] and the BC3 Corpus[67], originate from the W3C mailing list in 2004. Due to their origin, these emails are very technical in topic. We used the EEC due to its greater diversity of email subjects.

5.3.2 Gold Standard Thread Extraction from the Enron Email Corpus

As previously explained in Section 5.1, we define an email thread as a directed graph of emails connected by reply-to and forward relations. In this way, we attempt to identify email discussions between users. However, the precise definition of an email thread actually depends on the implementation that we, or any other researchers, used to identify the thread.

Previous researchers have derived email thread structure from a variety of sources. Wu and Oard (2005), and Zhu et al. (2005) auto-threaded all messages with identical, non-trivial, `Fwd:` and `Re:`-stripped `Subject` headers. Klimt and Yang (2004) auto-threaded messages that had stripped `Subject` headers and were among the same users (addresses). Lewis and Knowles (1997) assigned emails to threads by matching quotation structures between emails. Wan and McKeown (2004) reconstructed threads by header Message-ID information.

As the emails in the EEC do not contain any inherent thread structure, it was necessary for us to create email threads. First, we implemented Klimt and Yang (2004)'s technique of clustering the emails into threads that have the same `Subject` header (after it has been stripped of prefixes such as `Re:` and `Fwd:`) and shared participants. To determine whether emails were among the same users, we split a `Subject`-created email proto-thread apart into any necessary threads, such that the split threads had no senders or recipients (including `To`, `CC`, and `BCC`) in common.

The resulting email clusters had a number of problems. Clusters tended to over-group, because a single user included as a recipient for two different threads with the `Subject` "Monday Meeting" would cause the threads to be merged into a single cluster. In addition, many clusters consisted of all of the issues of a monthly subscription newsletter, or nearly identical peti-

[65] The EEC is in the public domain: `http://www.cs.cmu.edu/~enron/`

[66] `http://tides.umiacs.umd.edu/webtrec/trecent/parsed_w3c_corpus.html`

[67] `http://www.cs.ubc.ca/nest/lci/bc3.html`

[-]+ Auto forwarded by <anything > [-]+
[-]+ Begin forwarded message [-]+
[-]+ cc:Mail Forwarded [-]+
[-]+ Forwarded by <person > on <datetime > [-]+
[_]+ Forward Header [_]+
[-]+ Forwarded Letter [-]+
[-]+ Forwarded Message: [-]+
"<person > " wrote:
Starts with To:
Starts with <
... and more ...

Table 5.1: Representative examples of regular expressions for identifying quoted emails.

tions (see Klimt and Yang (2004)'s description of the "Demand Ken Lay Donate Proceeds from Enron Stock Sales" thread), or an auto-generated log of Enron computer network problems auto-emailed to the Enron employees in charge of the network. Such clusters of "broadcast" emails do not satisfy our goal of identifying email discussions between users.

Many email discussions between users exist in previously quoted emails auto-copied at the bottom of latter emails of the thread. A manual investigation of 465 previously quoted emails from 20 threads showed that none of them had interspersed comments or had otherwise been altered by more recent thread contributors. Threads in the Eec are quoted multiple times at various points in the conversation in multiple surviving emails. In order to avoid creating redundant threads, which would be an information leak risk during evaluation, we selected as the thread source the email from each Klimt and Yang (2004) cluster with the most quoted emails, and discarded all other emails in the cluster. This process is illustrated by Figure 5.17, which shows an original email thread, and an extracted email thread created by our method.

We used the quote-identifying regular expressions from Yeh and Harnly (2006) (see Table 5.1) to identify quoted previous emails.[68]

There are two important benefits to the creation methodology of the Etc[69]. First, since the emails were extracted from the same document, and the emails would only have been included in the same document by the email client if one was a `Reply` or `Forward` of the other, precision is very high (approaching 100%).[70] This is better precision than threads clustered from separate email documents, which may have the same `Subject`, etc. generating false positives. Some emails will inevitably be left out of the thread, reducing recall, because they were not part of the thread branch that was eventually used to represent the thread (as can be seen in Figure 5.17), or simply because they were not quoted. Our pairwise classification experiments, described in

[68]Some emails have no sender, etc., because they were only saved as incomplete drafts.

[69]We have made the Etc available online at `http://www.ukp.tu-darmstadt.de/data/text-similarity/email-disentanglement`

[70]In an analysis of 465 emails and 20 email threads, we found our system misidentified about 1% of emails. This was caused by regular expression mistakes.

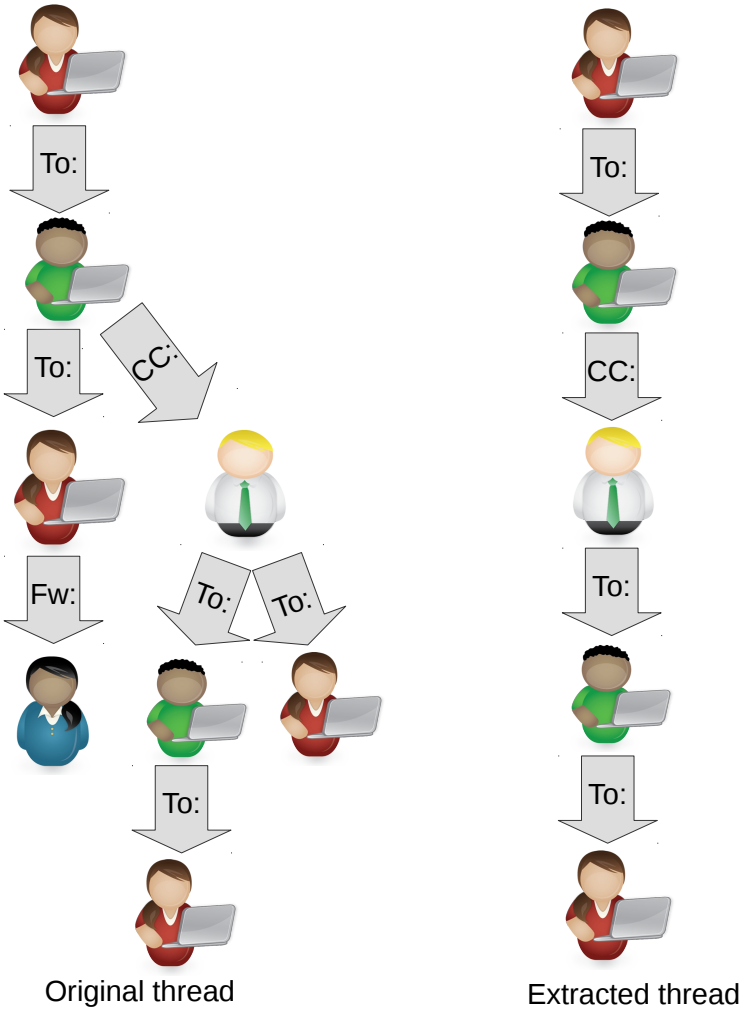

Figure 5.17: An original email thread, and an extracted email thread that has been created with our method. The longest branch of the original thread is used to produce the extracted email thread. To avoid the risk of partially identical threads, no other sequences are extracted.

Thread Size	Num threads
2	40,492
3	15,337
4	6,934
5	3,176
6	1,639
7	845
8	503
9	318
10	186
11-20	567
21+	181

Table 5.2: Thread sizes in the Etc.

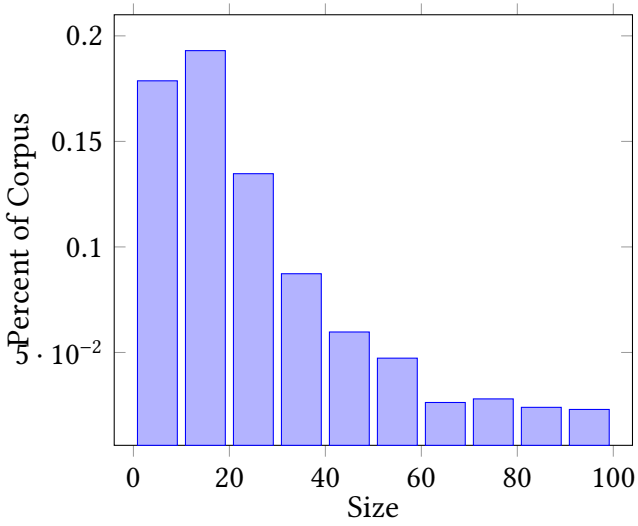

Figure 5.18: Percent of emails with a token count of 0-10, 10-20, etc.

Chapter 6, are unaffected by this reduced recall, because each experimental instance includes only a pair of emails, and not the entire thread.

Second, because the thread source did not require human annotation, using quoted emails gives us an unprecedented number of threads as data: 209,063 emails in 70,178 threads of two emails or larger. The sizes of email threads in the Etc is shown in Table 5.2. Email size by tokens as corpus percentage, for a 1,500 email pair sample, is shown in Figure 5.18. Emails have an average of 80.0±201.2 tokens, and an average count of 4.4±9.3 sentences.

Topic: "Grammatical Tense:gutted"

Turn1: *This article has been gutted. I deleted a lot of the cruft that had taken over, but a lot of former material is missing.[...]*

Turn2: *Good; the further this nest of doctrinaire obscurities is gutted, the better.*

Turn3: *Wait, you changed it to say that English doesn't have a future tense or you're citing that as an error (which it would naturally be)? For what it matters, [...]*

Turn4: *English doesn't have a future tense. It indicates the future with a modal (will) used with the present-tense inflection of the verb. [...]*

Figure 5.19: Excerpt from an EWDC discussion.

Discussion	# Turns	% Misind.	R	L	P(pos)
Grammatical_tense	20	.50	8	7	10/10
Hurricane_Iniki:1	15	.2	2	4	2/3
Hurricane_Iniki:2	13	.46	11	4	5/7
Possessive_adjective	13	.23	1	5	9/10
Prince's_Palace_of_Monaco	13	.54	9	9	6/6
Average	14.8	.39	6.2	5.8	.89

Table 5.3: Analysis of wrong indentation in 5 discussions, showing misindentation rate, the sum of how many tabs to the left or right are needed to fix the mis-indented response turn, and P of extracted positive pairs.

5.3.3 English Wikipedia Discussions Corpus

The English Wikipedia Discussions Corpus (EWDC) (Ferschke, 2014) is a corpus of discussions extracted from the discussion pages of Wikipedia. Wikipedia's discussion pages are used by Wikipedians to discuss proposed and implemented changes to Wikipedia articles. Each Wikipedia article has its own discussion page, and there may be many discussions listed on a discussion page. A sample discussion excerpt is shown in Figure 5.19[71].

Our EWDC dataset[72] consists of discussion turn pairs from the English Wikipedia Discussions Corpus. Discussion pages provide a forum for users to discuss edits to a Wikipedia article.

Adjacency in the EWDC is indicated by the user via text indent, as can be seen in Figure 5.19. Incorrect indentation (i.e., indentation that implies a reply-to relation with the wrong post) is common in longer discussions in the EWDC. In an analysis of 5 random threads longer than 10 turns each, shown in Table 5.3, we found that 29 of 74 total turns, or 39%±14pp of an average thread, had indentation that misidentified the turn to which they were a reply. We also found

[71] This figure can also be found in Section 7.1

[72] www.ukp.tu-darmstadt.de/data/wikidiscourse

that the misindentation existed in both directions: an approximately equal number of tabs and tab deletions were needed in each article to correct the mis-indented turns.

To minimize the number of turn pairs with incorrect indentation extracted from the corpus, we extracted our positive and negative pairs as follows: An adjacent pair is defined as a pair of turns in which one turn appears directly below the other in the text, and the latter turn is indented once beyond the previous turn. A non-adjacent pair is defined as a pair of turns in which the latter turn has fewer indents than the previous turn. Our extraction method yields 32 true positives and 4 false positives (precision = 0.89) in the 5 discussions. In an error analysis of 20 different pairs from our adjacency recognition experiments discussed in Chapter 7.7.2, we similarly found 0.90 class-averaged precision.

We derived a class-balanced dataset of 2684 pairs of adjacent and non-adjacent discussion turn pairs from the EWDC. The pairs came from 550 discussions within 83 Wikipedia articles. The average number of discussions per article was 6.6. The average number of extracted pairs per discussion was 4.9. The average turn contained 81±95 tokens (standard deviation) and 4±4 sentences. To reduce noise, usernames and time stamps have been replaced with generic strings.

5.4 Chapter Summary

In this chapter, we have provided background on several key aspects of discussion thread reconstruction. We have defined discussion threads, and provided examples of their many forms, as seen in Web 2.0, as well as problems related to these forms. We have reviewed research related to discussion threads and thread reconstruction. Finally, we have described the particular discussions corpora used for the experiments for this thesis, our Enron Threads Corpus (ETC) and the English Wikipedia Discussions Corpus (EWDC) (Ferschke, 2014). The ETC is a contribution of this thesis and is available online.

In the following chapters, we investigate thread reconstruction via a set of sequential subtasks. Chapter 6 investigates thread disentanglement, treating it as a pairwise text similarity classification problem, using email threads from the topically-diverse ETC. Chapter 7 investigates the recognition of adjacency pairs using lexical pairs, in the direct reply context of Wikipedia discussions. Chapter 8 investigates the recognition of adjacency pairs using the knowledge-rich technique of lexical expansion. These subtasks cover all the natural language processing necessary to extract pairwise turn information for further processing using graph-building techniques, a task which is outside the scope of this thesis.

Chapter 6

Email Thread Disentanglement

Thread reconstruction can be broken down into multiple, sequential sub-tasks. The input to the subtask sequence is an unordered bag of discussion turns. The output from the subtask sequence is a directed, rooted graph structure in which nodes represent discussion turns and edges represent asymmetric *reply-to* relations. We break down thread reconstruction into sequential steps: (1) thread disentanglement, to sort the turns according to their thread membership, (2) adjacency recognition, or identifying pairs of turns with reply-to relations, (3) graph construction, or building the best-fitting graph of the discussion. Because this thesis is concerned with the natural language aspects of thread reconstruction, we investigate steps (1) thread disentanglement, and (2) adjacency pair recognition. We leave aside (3) graph reconstruction as a topic of research in a different field.

In this chapter, we investigate thread disentanglement, as applied to emails. Specifically, we treat thread disentanglement as a pairwise classification problem, with the output of the classifier representing the relation between a pair of emails. In future work, such output will serve as the input for thread graph reconstruction, as weights of graph edges between nodes of individual emails. We address the following research questions:

> *Research Question*: Can text similarity be used for pairwise classification email thread disentanglement? Do content, style, structure, or semantic text similarity metrics perform best?
>
> *Research Question*: How does semantic similarity of the corpus affect the use of text similarity features for thread disentanglement?

The chapter is structured as follows. First, we provide an overview of our motivation (Section 6.1), with a discussion of previous research. We provide a description of the text similarity features, along with examples motivating their use, in Section 6.2. As a dataset, we use the Etc dataset described in Section 5.3.1. We describe our disentanglement experiments in Section 6.3, where each pair of emails is to be classified as *positive* (same thread) or

negative (different thread) instances, using text similarity features from each pair of emails. However, the effectiveness of text-similarity-based thread disentanglement may be influenced by the internal semantic similarity of the corpus, so we explore the effect that corpus topic distribution has on our email disentanglement task, by comparing several methods of sampling negative instances to avoid bias in the classifier. We use three different data samplings of negative instances: random class-balanced, semantically-matched class-balanced, and random class-imbalanced. We discuss inherent limitations of the task in Section 6.3.3, and error analysis in Section 6.3.4. We conclude the chapter with a summary of our findings in Section 6.4.

Most of the material in this chapter was previously published in peer-reviewed proceedings:

> **Emily K. Jamison** and Iryna Gurevych: 'Headerless, Quoteless, but not Hopeless? Using Pairwise Email Classification to Disentangle Email Threads', in: *Proceedings of 9th Conference on Recent Advances in Natural Language Processing* (RANLP 2013), Hissar, Bulgaria, 2013.

6.1 Motivation

When federal investigators seized the email servers from the Enron Corporation in 2001-2002, a new era was dawning: email forensics. Employees at Enron were suspected of having orchestrated extensive white collar crime within the company, but accountants had shredded most relevant company documents, so investigators turned to their emails, hoping the emails contained documentation of the crimes.

But investigators faced a problem that would become ubiquitous in electronic communication forensics in the coming decade: there were simply too many emails for investigators to process. How could the investigators extract the information they needed from email conversations, to gather evidence of the crimes? How could they even isolate the small percentage of emails most likely to contain evidence, that their agency had the manpower to process? Emails are frequently useless on their own; it is only within the context of the *email thread* that an email becomes meaningful.

Most modern emails contain useful metadata such as the MIME header `In-Reply-To`, which marks relations between emails in a thread and can be used to disentangle threads. However, email users sometimes attempt to obfuscate relevant email conversations to discourage investigation. In November 2012, the director of the CIA, former U.S. 4-star General David Petraeus was forced to resign his position over the scandal involving a cover up of emails: a special email account was opened, and the participants never actually sent emails from this account; they merely saved drafts of the emails, which eliminated many email headers. Law

enforcement described this trick of saving email drafts as "known to terrorists and teenagers alike".

Easy methods of obfuscating email threads include: opening an email account for a single purpose; using multiple email accounts for one person; sharing one email account among multiple persons; changing the `Subject` header; and removing quoted material from earlier in the thread.

How can emails be organized by thread without metadata such as their MIME headers?

We propose to use text similarity metrics to identify emails belonging to the same thread. In this chapter, as a first step for temporal thread disentanglement, we perform pairwise classification experiments on texts in emails using no MIME headers or quoted previous emails. We have found that content-based text similarity metrics outperform a Dice baseline, and that structural and style text similarity features do not; adding these latter feature groups does not significantly improve total performance. We also found that content-based features continue to outperform the others in both a class-balanced and class-imbalanced setting, as well as with semantically controlled or non-controlled negative instances.

In NLP, Elsner and Charniak (2010) described the task of *thread disentanglement* as "the clustering task of dividing a transcript into a set of distinct conversations," in which extrinsic thread delimitation is unavailable and the threads must be disentangled using only intrinsic information. In addition to emails with missing or incorrect MIME headers, entangled electronic conversations occur in environments such as interspersed internet relay chat conversations, web 2.0 article response conversations that do not have a hierarchical display order, and misplaced comments in Wiki Talk discussions.

Research on disentanglement of conversation threads has been done on internet relay chats (Elsner and Charniak, 2010), audio chats (Aoki et al., 2003), and emails *with* headers and quoted material (Yeh and Harnly, 2006; Erera and Carmel, 2008). However, to the best of our knowledge, no work has investigated reassembling email threads *without* the help of MIME headers or quoted previous emails.

6.2 Text Similarity Features

We investigate email thread disentanglement as a text similarity problem. Ideally, there exists a text similarity measure that marks pairs of emails from the same thread as *more similar* than pairs of emails from different threads. We evaluate a number of text similarity measures, divided according to Bär et al. (2011)'s three groups: Content Similarity, Structural Similarity, Style Similarity. Each set of features investigates a different manner in which email pairs from the same thread may be identified. In our experiments, all features are derived from the body of the email, while all headers such as `Recipients`, `Subject`, and `Timestamp` are ignored.

Content features Content similarity metrics capture the string overlap between emails with similar content. A pair of emails with a high content overlap is shown below. The emails have many tokens in common; this similarity will be quantified in various ways by the content features. Intuitively, emails from the same thread may repeatedly mention particular events, locations, names, dates, etc., which should be captured by the content features.

> **Email1**: *Please RSVP if you are attending the Directors Fund Equity Board Meeting next Wednesday, Nov 5, at 3pm.*
> **Email2**: *Yes, I'll be at the Directors Fund Equity Board Meeting on Wednesday, Nov 5, at 3pm.*

The *Longest Common Substring measure* (Gusfield, 1997) identifies uninterrupted common strings, while the *Longest Common Subsequence measure* (Allison and Dix, 1986) and the single-text-length-normalized *Longest Common Subsequence Norm measure* identify common strings containing interruptions and text replacements and *Greedy String Tiling measure* (Wise, 1996) allows reordering of the subsequences. Other measures which treat texts as sequences of characters and compute similarities with various metrics include *Levenshtein* (1966), *Monge Elkan Second String measure* (Monge and Elkan, 1997), *Jaro Second String* measure (Jaro, 1989), and *Jaro Winkler Second String* measure (Winkler, 1990). A *cosine similarity-type measure* was used, based on term frequency within the document. Sets of n-grams from the two emails are compared using the Jaccard coefficient (from Lyon et al. (2004)) and Broder's (1997) *Containment* measure.

Structural features Structural features attempt to identify similar syntactic patterns between the two texts, while overlooking topic-specific vocabulary. We propose that structural features, as well as style features below, may help in classification by means of communication accommodation theory (Giles and Ogay, 2007): speakers are known to adjust their speaking styles based on the language of other participants in the discussion. Structural features may identify syntactic accommodation between email discussion participants. Here we show a pair of emails with a high structural overlap, as measured by word pairs occurring in the same order and on part-of-speech n-gram similarity.

> **Email1**: *Can you attend the function next Wednesday, Nov 5, at 3pm?*
> **Email2**: *Can I attend the meeting? Sure!*

Stamatatos's *Stopword n-grams* (2011) capture syntactic similarities, by identifying text reuse where just the content words have been replaced and the stopwords remain the same. We measured the stopword n-gram overlap with Broder's (1997) *Containment* measure and four different stopword lists. We also tried the *Containment* measure and an *n-gram Jaccard measure* with *part-of-speech* tags. *Token Pair Order* (Hatzivassiloglou et al. 1999) uses pairs of words occurring in the same order for the two emails; *Token Pair Distance* (Hatzivassiloglou

et al., 1999) measures the distance between pairs of words. Both measures use computed feature vectors for both emails along all shared word pairs, and the vectors are compared with Pearson correlation.

Style features Style similarity reflects authorship attribution and surface-level statistical properties of texts. Intuitively, these features should identify accommodation between speakers that is not syntactic, as well as authorship, which may be helpful (although not sufficient) for thread disentanglement. Below, we show a pair of emails with high style similarity, contrasted against a pair of emails with low style similarity, as measured by the feature *Sentence length* (defined below).

High Style Similarity:

Email1: *Can you come next Wednesday? Fred Smith will explain the new project.*
Email2: *I am available this Wednesday. But I need to leave by 11.*

Low Style Similarity:

Email1: *Can you come next Wednesday? Fred Smith will explain the new project.*
Email2: *Yes, I'll be at the Directors Fund Equity Board Meeting on Wednesday, Nov 5, at 3pm. As you may be aware, Janet Brown will be hosting a Traders Reception after the meeting at the Blue Owl Ballroom, and we have been requested to escort Mr. Smith to the reception.*

Type Token Ratio (TTR) measure calculates text-length-sensitive and text-homogeneity-sensitive vocabulary richness (Templin, 1957). However, as this measure is sensitive to differences in document length between the pair of documents (documents become less lexically diverse as length and token count increases but type count levels off), and fluctuating lexical diversity as rhetorical strategies shift within a single document, we also used *Sequential TTR* (McCarthy and Jarvis, 2010), which corrects for these problems. *Sentence Length* and *Token Length* (inspired by (Yule, 1939)) measure the average number of tokens per sentence and characters per token, respectively. *Sentence Ratio* and *Token Ratio* compare *Sentence Length* and *Token Length* between the two emails (Bär et al., 2011). *Function Word Frequencies* is a Pearson's correlation between feature vectors of the frequencies of 70 pre-identified function words from Mosteller and Wallace (1964) across the two emails. We also compute *Case Combined Ratio*, showing the percentage of UPPERCASE characters in both emails combined ($\frac{UPPERCASE_{e1}+UPPERCASE_{e2}}{ALLCHARS_{e1}+ALLCHARS_{e2}}$), and *Case Document similarity*, showing the similarity between the percentage of UPPERCASE characters in one email versus the other email.

6.3 Evaluation

In this series of experiments, we evaluate the effectiveness of different feature groups to classify pairs of emails as being from the same thread (*positive*) or not (*negative*). Each instance to

be classified is represented by the features from a pair of emails and the instance classification, positive or negative.

We used a variation of K-fold cross-validation for evaluation. The 10 folds contained carefully distributed email pairs such that email pairs with emails from the same thread were never used in pairs of training, development, and testing sets, to avoid information leakage. Otherwise, it is possible that the classifier would learn patterns specific to the particular email threads instead of patterns generalizable to unseen data. All instances have been contained in exactly one test set. Instance division was roughly 80% training, 10% development, and 10% test data. Reported results are the weighted averages across all folds.

The evaluation used logistic regression, as implemented in Weka (Hall et al., 2009). Default parameters were used. Experiment preprocessing used the open-source DKPro Core (Eckart de Castilho and Gurevych, 2014), and most of the similarity measures are from the open-source DKPro Similarity (Bär et al., 2013). We use a baseline algorithm of Dice similarity between the texts of the two emails as a simple measure of token similarity. We created an upper bound by annotating 100 positive and 100 negative instances from the RB (class-balanced) dataset. A single native English speaker annotator answered the question, "Are these emails from the same thread?" Accuracy was found to be 89%, as shown in Table 6.1.

6.3.1 Data Sampling

Although we had 413,814 positive instances available in the Enron Threads Corpus, we found that classifier performance on a separate development set did not improve with additional training data, from about 200 training instances (see Figure 7.3). However, because the standard deviation in the data did not level out until around 1,200 class-balanced training instances[73], we used this number of positive instances (600) in each of our experiments.

In order to estimate effectiveness of features for different data distributions, we used three different subsampled datasets.

Random Balanced (RB) Dataset. The first dataset is class-balanced and uses 1200 training instances. Minimum email length is one word[74]. For every positive instance we used, we created a negative email pair by taking the first email from the positive pair and pseudo-randomly pairing it with another email from a different thread that was assigned to the same training, development, or test set. As explained in Chapter 5.3.2, due to construction techniques, the ETC dataset has very high thread precision, ensuring the gold labels of these pairs.

However, the probability of semantic similarity between two emails in a positive instance is much greater than the probability of semantic similarity between two emails in a randomly-created negative instance. The results of experiments on our first dataset reflect both the suc-

[73]Each fold used 1,200 training instances and 150 test instances.

[74]A one-token email example is "Thanks." While future work should investigate the effectiveness of text similarity features for very short emails, we include these emails as a small but existent part of the ETC dataset.

cess of our text similarity metrics and the semantic similarity (i.e., topical distribution) within our dataset. The topical distribution will vary immensely between different email corpora. To investigate the performance of our features in a more generalizable environment, we created a subsampled dataset that controls for semantic similarity within and outside of the email thread.

Semantically Balanced (SB) Dataset. The second dataset combines the same positive instances as the first set with an equal number of semantically-matched negative instances for a training size of 1200 instances, and a minimum email length of one word. For each positive instance, we measured the semantic similarity within the email pair using cosine similarity and then created a negative instance with the same (±.005) similarity. If a negative instance with the desired similarity could not be found in 1,600 tries, we used the closest match. The average similarity between emails in a negative instance was .44 while the average similarity between emails in a positive instance was .48[75]. Emails had an average of 96±287 tokens and 5±11sentences, and a similar token size distribution as RB.

Random Imbalanced (RI) Dataset. However, both the RB and SB datasets use a class-balanced distribution. To see if our features are still effective in a class-imbalanced environment, we created a third dataset with a 90% negative, 10% positive distribution for both the training and test sets[76]. To create this dataset, we augmented RB with an extra 8 negative instances for each positive instance. Experiments with this dataset use 10-fold cross-validation, where each fold has 6000 training and 750 test instances. No minimum email length was used, similar to a more natural distribution.

6.3.2 Results

Our results are shown in Table 6.1. System performance is reported in $F_1=\frac{2\times P(pos)\times R(pos)}{P(pos)+R(pos)}$ and Accuracy=$\frac{TP+TN}{TP+FN+TN+FP}$. F_1 is measured on the positive class (i.e., pairs of emails from the same thread): because the natural class-imbalance inherent to the Etc makes positive pairs very rare, we focus on their detection. Accuracy is included to show performance on both positive and negative classes. Feature groups are shown in isolation as well as ablation (i.e., the complete set of features minus one group). [77]

With the RB corpus, the best performing single feature configuration, content features group (P=.83 ±.04), matches the human upper bound (described in Section 6.3.4) precision

[75]Because some pairs will exceed the 1,600 attempt threshold and have a greater-than-1-percentage-point difference, a series of experiments found that a class divide of 4 percentage points was as close as we could reasonably get. With 1,600 attempts and 1-percentage-point cutoff, 17% reached the attempt limit.

[76]This class imbalance is still artificially lower than a more natural 99.99+% negative natural class imbalance.

[77]Additionally, we tried a semantic similarity measures feature group. We used Gabrilovich & Markovitch's (2007) *Explicit Semantic Analysis* (ESA) vector space model, with vectors from Bär et al. (2011) using three different lexical-semantic resources: WordNet, Wikipedia, and Wiktionary. The performance of this feature group (P=.50) was not good enough to include in Table 6.1.

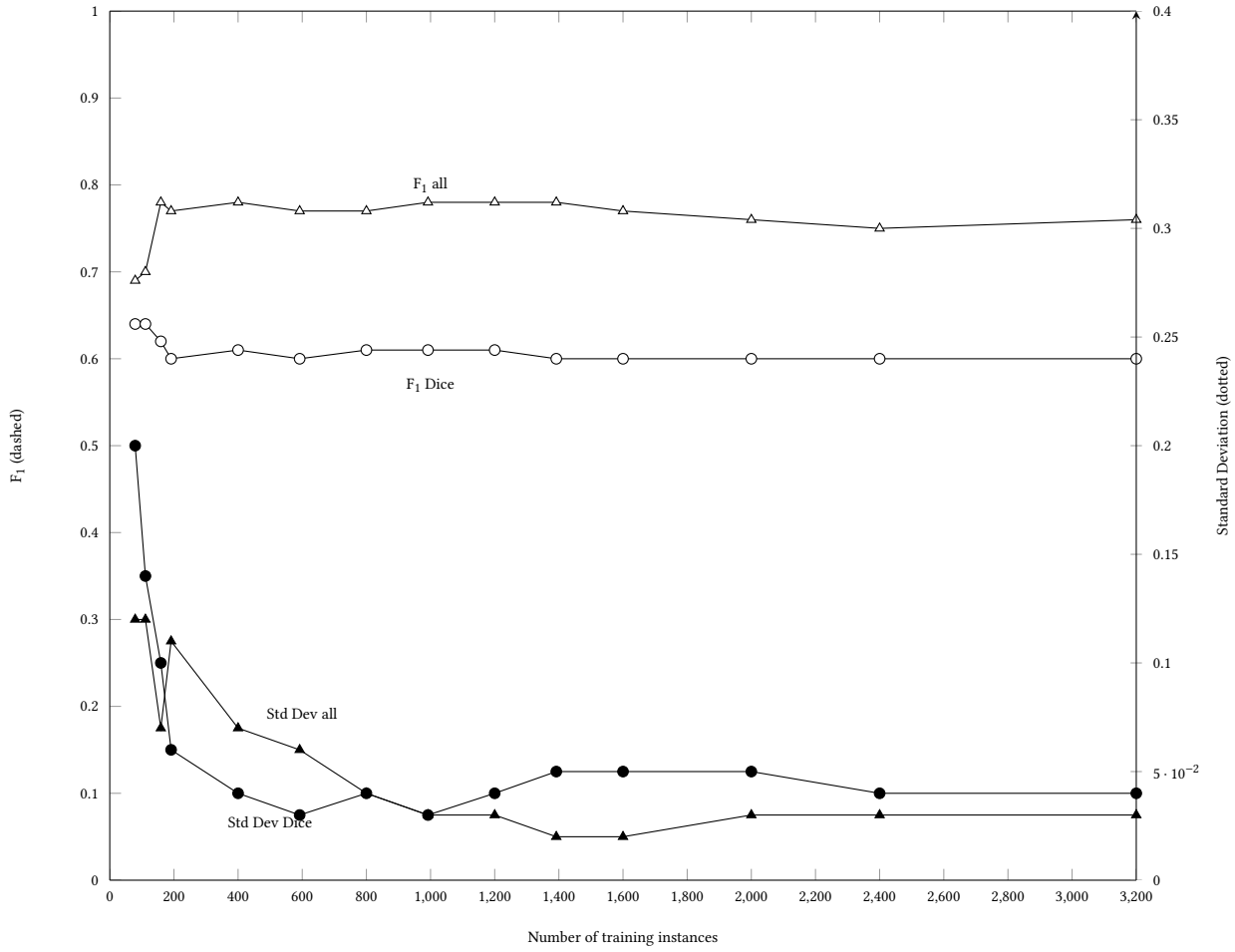

Figure 6.1: Training data sizes and corresponding F_1 and standard deviation.

Feature	RB F_1	SB F_1	RI F_1	RB Acc	SB Acc	RI Acc
Chance	.50	.50	.90	.50	.50	.90
Dice Baseline	.61 ±.04	.56 ±.04	.09 ±.04	.63 ±.03	.58 ±.03	.9 ±.0
Human upper bound	.89	-	-	.89	-	-
Just content	.78 ±.03	.65 ±.04	.38 ±.06	.79 ±.03	.67 ±.03	.92 ±.01
Just struct	.42 ±.05	.33 ±.04	.06 ±.05	.55 ±.03	.52 ±.03	.90 ±.00
Just style	.60 ±.05	.57 ±.03	.00 ±.00	.60 ±.04	.56 ±.03	.90 ±.00
No content	.60 ±.03	.55 ±.03	.08 ±.05	.62 ±.03	.57 ±.02	.90 ±.00
No struct	.78 ±.03	.66 ±.03	.41 ±.06	.79 ±.02	.67 ±.02	.92 ±.01
No style	.78 ±.03	.63 ±.04	.38 ±.06	.79 ±.03	.65 ±.03	.92 ±.00
Everything	.78 ±.02	.65 ±.03	.40 ±.05	.79 ±.02	.67 ±.03	.92 ±.00

Table 6.1: Email pair classification results. Standard deviation is reported from the variance in the CV results.

(P=.84). The benefit of content features is confirmed by the reductions in complete feature set performance when they are left out. The content features group was the only group to perform significantly above the Dice baseline. Adding the other feature groups does not significantly improve the overall results. Further leave-one-out experiments revealed no single high performing feature within the content features group.

Structural features produced low performance, failing to beat the Chance baseline. As a rhetorical strategy, syntactic repetition as indicated by structural similarity is rare in an email conversational setting. Any structural benefits are likely to come from sources unavailable in a disguised email situation, such as auto-signatures identifying senders as the same person. The low results on structural features show that we are not relying on such artifacts for classification.

Style features were also unhelpful, failing to significantly beat the Dice baseline. The features failed to identify communication accommodation within the thread.

Results on the SB dataset show that there is a noticeable drop in classification for all feature groups when negative instances have a similar semantic similarity as positive instances. The configuration with all features showed a 15 percentage point drop in precision, and a 12 percentage point drop in accuracy. However, content features continues to be the best performing feature group with semantically similar negative instances, as with random negative instances, and outperformed the Dice baseline. Adding the additional feature groups does not significantly improve overall performance.

The results on the imbalanced (RI) corpus mirror results from the balanced (RB) corpus. The best-performing individual feature group in both experiments was the content feature group; in the class-imbalanced experiments the group alone beats the Dice baseline in F_1 by 29 percentage points and reduces accuracy error by about 20%.

Elsner and Charniak (2011) use coherence models to disentangle chat, using some features (entity grid, topical entity grid) which correspond to the information in our content features group. They also found these content-based features to be helpful.

Text	Freq in Corpus
FYI	48
FYI <*name* >	23
one person's autosignature	7
Thanks!	5
Please print.	5
yes	4
FYI, Kim.	3
ok	3
please handle	3

Table 6.2: Common entire email texts and their frequencies in the corpus.

6.3.3 Inherent limitations

Certain limitations are inherent in email thread disentanglement. Some email thread relations cannot be detected with text similarity metrics, and require extensive discourse knowledge. In the emails below, discourse knowledge is needed to resolve *there* with the event the Directors Fund Equity Board Meeting.

> **Email1**: *Can you attend the Directors Fund Equity Board Meeting next Wednesday, Nov 5, at 3pm?*
> **Email2**: *Yes, I will be there.*

Several other problems in email thread disentanglement cannot be solved with any discourse knowledge. One problem is that some emails are identical or near-identical; there is no way to choose between textually identical emails. Table 6.2 shows some of the most common email texts[78] in our corpus.

However, near identical texts make up only a small portion of the emails in our corpus. In a sample of 5,296 emails, only 3.6% of email texts were within a .05 Jaro Second String similarity value of another text.

Another problem is that some emails are impossible to distinguish without world and domain knowledge. Consider a building with two meeting rooms: *A101* and *A201*. Sometimes *A101* is used, and sometimes *A201* is used. In response to the question, *Which room is Monday's meeting in?*, there may be no way to choose between *A101* and *A201* without further world knowledge.

Another problem is topic overlap. For example, in a business email corpus such as the EEC, there are numerous threads discussing Monday morning 9am meetings. The more similar the

[78]Email texts were recognized as identical to each other if their Jaro Second String similarity was <.05, allowing matching of emails that differed by a few characters.

language used between threads, the more difficult the disentanglement becomes, using text similarity. This issue is addressed with the SB dataset.

Finally, our classifier cannot out-perform humans on the same task, so it is important to note human limitations in email disentanglement. Our human upper bound is shown in Table 6.1. We will further address this issue in Sections 6.3.4.

6.3.4 Error Analysis

We inspected 50 email pairs each of true positives, false positives, false negatives, and true negatives from our RB experiments[79] . We inspected for both technical details likely to affect classification, and for linguistic features to guide future research. Technical details included small and large text errors (such as unidentified email headers or incorrect email segmentation), custom and non-custom email signatures, and the presence of large signatures likely to affect classification. Linguistic features included an appearance of consecutivity (emails appear in a Q/A relation, or one is informative and one is 'please print', etc.), similarity of social style ("Language vocab level, professionalism, and social address are a reasonable match"), and the annotator's perception that the emails could be from the same thread.

An example of a text error is shown below.

Sample text error:
Craig Young
09/08/2000 01:06 PM

Names and dates occur frequently in legitimate email text, such as meeting attendance lists, etc., which makes them difficult to screen out. Emails from false positives were less likely to contain these small errors (3% versus 14%), which implies that the noise introduced from the extra text has more impact than the false similarity potentially generated by similar text errors. Large text errors (such as 2 emails labeled as one) occurred in only 1% of emails and were too rare to correlate with results.

Auto signatures, such as the examples below, mildly impacted classification.

Custom Auto signature:
Carolyn M. Campbell
King& Spalding
713-276-7307 (phone)
713-751-3280 (fax)
ccampbell@kslaw.com <mailto:ccampbell@kslaw.com>

[79]Despite the semantic similarity control, an error analysis of our SB experiments showed no particularly different results.

Non-custom Auto signature:
Get your FREE download of MSN Explorer
at http://explorer.msn.com

Instances classified as negative (both FN and TN) were marginally more likely to have had one email with a non-customized auto signature (3% versus 1.5%) or a customized auto-signature (6.5% versus 3.5%). Auto signatures were also judged likely to affect similarity values more often on instances classified as negative (20% of instances). The presence of the auto signature may have introduced enough noise for the classifier to decide the emails were not similar enough to be from the same thread. We define a non-custom auto-signature as any automatically-added text at the bottom of the email. We did not see enough instances where both emails had an auto signature to evaluate whether similarities in auto signatures (such as a common area code) impacted results.

Some email pair similarities, observable by humans, are not being captured by our text similarity features. Nearly all (98%) positive instances were recognized by the annotator as potential consecutive emails within a thread, or non-consecutive emails but still from the same thread, whereas only 46% of negative instances were similarly (falsely) noted. Only 2% of negative instances were judged to look like they were consecutive emails within the same thread.

The following TP instance shows emails that look like they could be from the same thread but do not look consecutive.

Email1: *give me the explanations and i will think about it*
Email2: *what do you mean, you are worth it for one day*

Below is a TN instance with emails that look like they could be from the same thread but do not look consecutive.

Email1: *i do but i havent heard from you either, how are things with wade*
Email2: *rumor has it that a press conference will take place at 4:00 - more money in, lower conversion rate.*

The level of professionalism ("Language vocab level, professionalism, and social address are a reasonable match") was also notable between class categories. All TP instances were judged to have a professionalism match, as well as 94% of FN's. However, only 64% of FP's and 56% of TN's were judged to have a professionalism match. Based on a review of our misclassified instances, we are surprised that our classifier did not learn a better model based on style features (F_1=.60). Participants in an email thread appear to echo the style of emails they reply to. For instance, short, casual, all-lowercase emails are frequently responded to in a similar manner.

6.4 Chapter Summary

In this chapter, we have investigated the use of text similarity features for the pairwise classification of emails for thread disentanglement. Automatic thread disentanglement is an important step in thread reconstruction.

We answered the following questions, posed at the beginning of this chapter:

> *Research Question*: Can text similarity be used for pairwise classification email thread disentanglement? Do content, style, structure, or semantic text similarity metrics perform best?

We have found that content similarity features are more effective than style or structural features, and we have found that semantic features are ineffective, perhaps due to the domain-specific nature of emails. There appear to be more stylistic features uncaptured by our similarity metrics, which humans access for performing the same task.

> *Research Question*: How does semantic similarity of the corpus affect the use of text similarity features for thread disentanglement?

We have shown that semantic differences between corpora will impact the general effectiveness of text similarity features, but that content features remain effective.

In the next two chapters (Chapters 7 and 8), we explore automatic adjacency recognition. In Chapter 7, we investigate adjacency recognition using lexical pairs, a statistical/distributional feature technique. In Chapter 8, we investigate adjacency recognition using lexical expansion via human-compiled lexical semantic resources, a knowledge-rich feature technique. The contributions of this chapter, when combined with the next two chapters, provide the natural language processing foundation for thread reconstruction.

Chapter 7

Wikipedia Discussion Adjacency Pairs

A critical step in thread reconstruction is *adjacency recognition*, or the recognition of reply-to relations between pairs of discussion turns. When adjacent pairs of turns can be recognized accurately, the classifier results can be used to create a directed graph of the discussion.

In this thesis, we have proposed that thread reconstruction consists of three sequential steps: (1) thread disentanglement, to sort the turns according to their thread membership, (2) adjacency recognition, or identifying pairs of turns with reply-to relations, (3) graph construction, or building the best-fitting graph of the discussion. In the previous chapter (Chapter 6), we performed automatic thread disentanglement (step 1) as pairwise classification by using features that measured various forms of text similarity between the pairs.

In this chapter, we investigate the recognition of adjacency pairs, as applied to Wikipedia discussion turns. Wikipedia discussion turns are better suited to adjacency recognition than email threads because all responses to an earlier message are replies, while email responses may be forwarded messages, etc., or otherwise lack discussion structure. We approach this task as a pair classification task, and we propose features that are particularly suited for the pair classification paradigm. We address the following research questions:

> *Research Question*: Are lexical pairs of discourse connectives, stopwords, unigrams, or bigrams effective for adjacency recognition? Does adding discourse information or removing stopwords or adding feature symmetry help?
>
> *Research Question*: Is topic bias inflating these results?

The chapter is structured as follows. First, we provide an overview of our motivation (Section 7.1), background information on adjacency pair typologies (Section 7.2), and a discussion of previous research (Section 7.3). We describe our human performance annotation experiment, which was used to determine an upper bound for this task on our dataset (Section 7.5).

We describe and motivate our feature sets (Section 7.6). Our first set of automatic adjacency recognition experiments used no topic bias control, and is described in Section 7.7. The problem of topic bias and solutions for its control are discussed in Section 7.8. We re-run our experiments using topic bias control, and compare the results with our non-topic-bias-controlled experiments in Section 7.9. We conclude the chapter in Section 7.10 with a summary of our findings.

Most of the material in this chapter was previously published in peer-reviewed proceedings:

Emily K. Jamison and Iryna Gurevych: 'Adjacency Pair Recognition in Wikipedia Discussions using Lexical Pairs', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing* (PACLIC), Phuket, Thailand, 2014.

7.1 Motivation

A growing cache of online information is contained inside user-posted forum discussions. Thread structure of the discussion is useful in extracting information from threads: Wang et al. (2013) use thread structure to improve IR over threads, and Cong et al. (2008) use thread structure to extract question-answer pairs from forums. However, as Seo et al. (2009) point out, thread structure is unavailable in many forums, partly due to the popularity of forum software phpBB[80] and vBulletin[81], whose default view is non-threaded.

Thread reconstruction provides thread structure to forum discussions whose original thread structure is nonexistent or malformed, by sorting and re-ordering turns into a directed graph of adjacency (reply-to) relations. Pairs of adjacent turns (*adjacency pairs*) were first identified by Sacks et al. (1974) as the structural foundation of a discussion, and recognition of adjacency pairs is a critical step in thread reconstruction (Balali et al., 2014; Wang et al., 2008; Aumayr et al., 2011).

Figure 7.1 shows an excerpt from Ferschke's (2014) English Wikipedia Discussions Corpus (EWDC). Thread structure is indicated by text indents. Turn pairs *(1,2)*, *(1,3)*, and *(3,4)* are adjacency pairs; pairs *(2,3)* and *(1,4)* are not. Adjacency recognition is the classification of a pair of turns as adjacent or nonadjacent.

Although most previous work on thread reconstruction takes advantage of metadata such as user id, timestamp, and quoted material (Aumayr et al., 2011; Wang et al., 2011a), metadata is unreliable in some forums, such as Wikipedia Discussion page forums, where metadata and user contribution are difficult to align (Ferschke et al., 2012). Wang et al. (2011c) find that joint prediction of dialogue act labels and adjacency recognition improves accuracy when compared

[80] http://www.phpbb.com/
[81] http://www.vbulletin.com/

Turn1: *This article has been gutted. I deleted a lot of the cruft that had taken over, but a lot of former material is missing.[...]*

Turn2: *Good; the further this nest of doctrinaire obscurities is gutted, the better.*

Turn3: *Wait, you changed it to say that English doesn't have a future tense or you're citing that as an error (which it would naturally be)? For what it matters, [...]*

Turn4: *English doesn't have a future tense. It indicates the future with a modal (will) used with the present-tense inflection of the verb. [...]*

Figure 7.1: Excerpt from the EWDC discussion *Grammatical Tense:gutted*.

to separate classification; dialogue act classification does not require metadata. However, existing dialogue act typologies are inapplicable for some forums (see Section 7.2.2).

In this chapter, we perform adjacency recognition on pairs of turns extracted from the English Wikipedia Discussions Corpus (EWDC). We use lexical pair features, which require neither metadata nor development of a dialogue act typology appropriate for Wikipedia discussions. We perform two sets of supervised learner experiments. First, we use lexical pairs for adjacency recognition in k-fold cross validation (CV) setting. Then we show how this permits topic bias, inflating results. Second, we repeat our first set of experiments, but in a special CV setting that removes topic bias. We find that lexical pairs outperform a cosine similarity baseline and a most frequent class baseline both without and with controlling for topic bias, and also exceed the performance of lexical strings of stopwords and discourse connectives on the task.

7.2 Background

Adjacency pairs were proposed as a theoretical foundation of discourse structure by Sacks et al. (1974), who observed that conversations are structured in a manner where the current speaker uses structural techniques to select the next speaker, and this structure is the adjacency pair: a pair of adjacent discussion turns, each from different speakers, and the relation between them.

7.2.1 Adjacency Pair Typologies

Previous work on adjacency recognition has found adjacency pair typologies to be useful (Wang et al., 2011c). Early work on adjacency pair typologies labeled adjacency pairs by adjacency relation function. Schegloff and Sacks (1973) proposed initial sequences (e.g., greeting exchanges), preclosings, pre-topic closing offerings, and ending sequences (i.e., terminal exchanges). Other adjacency pair typologies consist of pairs of dialogue act labels. Based on their work with transcripts of phone conversations, Sacks et al. (1974) suggested a few types of adjacency pairs: greeting-greeting, invitation-acceptance/decline, complaint-denial, compliment-

rejection, challenge-rejection, request-grant, offer-accept/reject, question-answer. In transcribed phone dialogues on topics of appointment scheduling, travel planning, and remote PC maintenance, Midgley et al. (2009) identified adjacency pair labels as frequently co-occurring pairs of dialog acts, including suggest-accept, bye-bye, request/clarify-clarify, suggest-reject, etc.

7.2.2 Discussion Structure Variation

Much adjacency pair descriptive work was based on transcriptions of phone conversations. Sacks et al. (1974) were discussing phone conversations when they observed that a speaker can select the next speaker by the use of adjacency pairs, and the subsequent speaker is obligated to give a response appropriate to and limited by the adjacency pair, such as answering a question. In a phone conversation, the participant set is fixed, and rules of the conversation permit the speaker to address other participants directly, and obligate a response.

However, in other types of discussion, such as forum discussions, this is not the case. For example, in QA-style forums such as CNET (Kim et al., 2010), a user posts a question, and anyone in the community may respond; the user cannot select a certain participant as the next speaker. Wikipedia discussions vary even further from phone conversations: many threads are initiated by users interested in determining community opinion on a topic, who avoid asking direct questions. Wikipedia turns that might have required direct replies from a particular participant in a speaker-selecting (SS) phone conversation, are formulated to reduce or remove obligation of response in this non-speaker-selecting context. Some examples are below; NSS turns are actual turns from the Ewdc.

Rephrasing a user-directed command as a general statement:

SS turn: "Please don't edit this article, because you don't understand the concepts."

NSS turn: "Sorry, but anyone who argues that a language doesn't express tense [...] obviously doesn't understand the concept of tense enough to be editing an article on it."

Obtaining opinions by describing past user action instead of questioning:

SS turn: "Which parts of this article should we delete?"

NSS turn: "This article has been gutted. I deleted a lot [...]."

Using a proposal instead of a question:

SS turn: "Should we rename this article?"

NSS turn: "I propose renaming this article to [...]"

Following questions with statements that deflect need for the question to be answered:

NSS turn: "Wait, you changed it to say that English doesn't have a future tense or you're citing that as an error (which it would naturally be)? For what it matters, even with the changes, this entire article needs a rewrite from scratch because so much of it is wrong."

Avoiding questions to introduce a new topic:
SS turn: "Have you heard of Flickr?"
NSS turn: "I don't know whether you know about Flickr or not, but theres a bunch of creative commons licensed images here some better and some worse than the article which you might find useful[...]".
Anticipating responses:
NSS turn: "What are the image names? :Image:Palazzo Monac.jpg has a problem, it's licensed with "no derivative works" which won't work on Commons.[...] If you meant other ones, let me know their names, ok?"

As seen above, Wikipedia discussions have different dialogue structure than phone conversations. Because of the different dialogue structure, existing adjacency pair typologies developed for phone conversations are not appropriate for Wikipedia discussions. As it would require much effort to develop an appropriate adjacency-pair typology for Wikipedia discussions, our research investigates the cheaper alternative of using lexical pairs to recognize adjacency pairs.

7.3 Related Work

To the best of our knowledge, our work is the first work that uses lexical pairs to recognize adjacency pairs.

7.3.1 Adjacency Recognition

Most previous work on thread reconstruction has, in addition to using metadata-based features, used word similarity, such as cosine similarity or semantic lexical chaining, between turn pairs for adjacency recognition or thread structure graph construction. Wang and Rosé (2010) trained a ranking classifier to identify "initiation-response" pairs consisting of quoted material and the responding text in Usenet `alt.politics.usa` messages, based on text similarity features (cosine, LSA). Aumayr et al. (2011) reconstructed discussion thread graphs using cosine similarity between pairs of turns, as well as reply distance, time difference, quotes, and thread length. They first learned a pairwise classification model over a class-balanced set of turn pairs, and then used the predicted classifications to construct graphs of the thread structure of discussions from the Irish forum site `Boards.ie`. Wang et al. (2011a) also reconstructed thread graphs using cosine similarity in addition to features based on turn position, timestamps, and authorship, using forum discussions from Apple Discussion, Google Earth, and CNET. Wang et al. (2008) reconstructed discussion threads of player chats from the educational legislative game *LegSim*, using tf-idf vector space model similarity between pairs of turns to build the graphs. Balali et al. (2014) included a feature of tf-idf vector-space model

of text similarity between a turn and a combined text of all comments, a feature of text similarity between pairs of turns, and an authorship language model similarity feature, to learn a pairwise ranking classifier, and then constructed graphs of the thread structures of news forum discussions. Wang et al. (2011d) evaluated the use of WordNet, Roget's Thesaurus, and WORDSPACE Semantic Vector lexical chainers for detecting semantic similarity between two turns and their titles, to identify thread-linking structure. Wang et al. (2011c) used a dependency parser, based on unweighted cosine similarity of titles and turn contents, as well as authorship and structural features, to learn a model for joint classification of Dialogue Acts and "inter-post links" between posts in the CNET forum dataset.

7.3.2 Lexical Pairs

We use lexical pairs as features for adjacency recognition. Although not previously used for this task, lexical pairs have been helpful for other discourse structure tasks such as recognizing discourse relations. Marcu and Echihabi (2002) used lexical pairs from all words, nouns, verbs, and cue-phrases, to recognize discourse relations. A binary relation/non-relation classifier achieves 0.64 to 0.76 accuracy against a 0.50 baseline, over approx. 1M instances. Lin et al. (2009b) performed discourse relation recognition using lexical pairs as well as constituent and dependency information of relations in the Penn Discourse Treebank. They achieved 0.328 accuracy against a 0.261 most frequent class baseline, using 13,366 instances. Pitler et al. (2009) performed binary discourse relation prediction using lexical pairs, verb information, and linguistically-motivated features, and achieve improvements of up to 0.60-0.62 accuracy, compared with a 0.50 baseline, on datasets sized 1,460 to 12,712 instances from the Penn Discourse Treebank. Biran and McKeown (2013) aggregated lexical pairs as clusters, to combat the feature sparsity problem. While improvements are modest, lexical pairs are helpful in these discourse tasks where useful linguistically-motivated features have proven elusive.

7.4 Dataset

We use the Ewdc dataset, which is described in detail in Section 5.3.3. This dataset consists of 2684 pairs of adjacent and non-adjacent discussion turns from the Ewdc Ferschke (2014).

7.5 Human Performance

We annotated a subset of our data, to determine a human upper bound for adjacency recognition. Two annotators classified 128 potential adjacency pairs (23 positive, 105 negative) in 4 threads with an average length of 6 turns. The annotators could see all other turns in the conversation, unordered, along with the pair in question. This pairwise binary classification scenario matches the pairwise binary classification in the experiments in Sections 7.7 and 7.9.

Each pair was decided independently of other pairs. Cohen's kappa agreement (Cohen, 1960) between the annotators was 0.63.

We noticed a common pattern of disagreement in two particular situations. When an "I agree" turn referred back to an adjacency pair in which one turn elaborated on the other, it was difficult for an annotator to determine which member of the original adjacency pair was the parent of the "I agree" comment. In a different situation, sometimes a participant contributed a substantially off-topic post that spawned a new discussion. It was difficult for the annotators to determine whether the off-topic post was a vague response to an existing post, or whether the off-topic post was truly the beginning of a brand-new discussion, albeit using the same original discussion thread.

7.6 Features

We use three types of features for adjacency recognition: lexical pairs, structural context information, and pair symmetry.[82]

Lexical pairs. A *lexical pair* feature consists of a pair of n-grams with one n-gram taken from the first document and one n-gram taken from the second document. An n-gram is a string of consecutive tokens of length n in a text. Following Marcu and Echihabi (2002), we find a relation (in our case, adjacency) that holds between two text spans, N_1, N_2, is determined by the n-gram pairs in the Cartesian product defined over the words in the two text spans $(n_i, n_j) \in N_1 \times N_2$.

The goal of using lexical pairs is to identify word pairs indicative of adjacency, such as (*why, because*) and (*?, yes*). These pairs cannot be identified using text similarity techniques used in previous work (Wang and Rosé, 2010).

In addition to lexical pairs created from document n-grams, lexical pairs were created from a list of 50 stopwords (Stamatatos, 2011), Penn Discourse Treebank discourse connectives (Prasad et al., 2008), and a particularly effective combination of just 3 stopwords: *and*, *as*, *for*. Other variables included the parameter n-gram n, and removed stopwords, which skipped unallowed words in the text.

Structural context information. Some of our feature groups include *structural context information* of the discussion turn codified as lexical items in the lexical pair string. We include sentence boundaries (SB), commas (CA), and sentence location (i.e., sentence occurs in first quarter, last quarter, or middle of the discussion turn). A sample lexical string representing text from the beginning of a turn is below.

[82]Because our goal is adjacency recognition based on text content features, we do not use indentation offset as a feature.

Text: *No, that is correct.*

Lexical string: `no-that-is-correct`

with struct.: `no-CA-that-is-correct-SBBEGIN`

Pair symmetry. Our dataset of discussion turn pairs retains the original order from the discussion. This permits us to detect order-sensitive features such as *(why, because)* and not *(because, why)*, in which the n-gram from Turn1 always occurs on the left-hand side of the feature name. Adjacency pairs, by definition, are nonsymmetrical. To confirm this property, in some of our feature groups, we extract a reverse-ordered feature for each standard feature. An example with *symmetrical* and *non-symmetrical* features is shown below.

Turn1: *Why ?*

Turn2: *Because .*

Non-Sym features: `(why, because)`

Sym features: `(why, because)`, `(because, why)`

7.7 Experiments without Topic Bias Control

In our first set of experiments, we perform adjacency recognition without topic bias control ("non-TBC"). We use the SVM classifier SMO (Hall et al., 2009) in the DKPro Text Classification (DKPro TC) framework (Daxenberger et al., 2014) for pairwise classification[83] and 5-fold[84] cross-validation (*CV*), in which all instances are randomly assigned to CV folds. These experiments do not control for any topic bias in the data. Previous work (Wang and Rosé, 2010) has structured adjacency recognition as a ranking task, with the classifier choosing between one correct and one incorrect response to a given turn. In our experiments, we use pairwise binary classification, because the high indentation error rate and our EWDC instance selection method did not yield enough matched turn pairs for ranking. Feature parameters (such as top k n-grams, string lengths, and feature combinations) were tuned using CV on a development subset of 552 pairs, while the final results reflect experiments on the remaining dataset of 2684 pairs. Results are shown as F-measure for class c=adjacent, nonadjacent): $F_{1_c}=\frac{2 \cdot P_c \cdot R_c}{P_c+R_c}$, and Accuracy=$\frac{TP+TN}{TP+FP+TN+FN}$. The most frequent class (*MFC*) baseline chooses the most frequent class observed in the training data, as calculated directly from the experiment. The cosine similarity (*CosineSim*) baseline is an SVM classifier trained over cosine similarity scores of the turn pairs. The `Human Upper Bound` shows agreement from Section 7.5 and reflects a natural limit on task performance.

[83] Although discourse turns are sequential, we classify individual pairs. Future work may investigate this as a sequence labeling task.

[84] Although 10-fold CV is more common in many NLP experiments, we use 5-fold cross validation in Section 7.7 to make our results directly comparable with results in Section 7.9.

Name	Words	N-gram Length	Context	Symmetry	removed words	F1+	F1-	Acc
Chance								.50
MFC						.44	.54	.49±.01
CosineSim						.62	.49	.56±.01
Human Upper Bound						.70	.93	.89
Stopwords+SB+NoSym	stopwords	1-3	SB	-	-	.61	.63	.62±.01
Stopwords+SB+Sym	stopwords	1-3	SB	Sym	-	.54	.56	.55±.02
Stopwords+noSB+noSym	stopwords	1-3	-	-	-	.57	.63	.60±.01
Stopwords+SB+CA+SL+noSym	stopwords	1-3	SB,CA,SL	-	-	.60	.63	.61±.01
DiscConn+SB+noSym	disc. conn.'s	1-3	SB	-	-	.60	.63	.61±.01
And-as-for	"and", "as", "for"	1-3	-	Sym	-	.63	.39	.54±.03
Pair1grams	all words	1	-	-	-	.62	.66	.64±.01
Pair2grams	all words	2	-	-	-	.60	.53	.57±.03
Pair1ngrams+noDC	all words	1	-	-	disc. conn.'s	.64	.66	.65±.02
pair1ngrams+noSW	all words	1	-	-	stopwords	.66	.70	**.68±.02**

Table 7.1: Non-TBC adjacency recognition feature set descriptions and results. F_1 results are shown by adjacent (+) and nonadjacent (-) classes. Accuracy is shown with cross-validation fold standard deviation. `Human Upper Bound` is calculated on a different dataset, which was also derived from the EWDC.

7.7.1 Results

Table 7.1 shows our feature combinations and results. All experiment combinations were $p \leqslant 0.05$ significantly different (McNemar, 1947) from the `CosineSim` and `MFC` baselines. The highest performing feature combination was pair unigrams with stopwords removed (`pair1grams+noSW`), which had higher accuracy (.68±.02) than all other feature combinations, including `pair1grams` that included stopwords (.64±.01), and all of the stopword feature sets. Stopword removal increases the system performance for our task, which is unexpected because in other work on different discourse relation tasks, the removal of stopwords from lexical pairs has hurt system performance (Blair-Goldensohn et al., 2007; Marcu and Echihabi, 2002; Biran and McKeown, 2013).

Longer n-grams did not increase performance: `pair2grams` (.57±.03) significantly underperformed `pair1grams` (.64±.01).

We examined the performance curve using various n numbers of most frequent lexical pairs as features on a subset of our corpus (1,380 instances). We found that there was no sharp benefit from a few particularly useful pairs, but that performance continued to increase as n approached 5000.

We found that the classifier performs better when the model learns turn pair order, and the reduced data sparsity from using symmetrical features was not valuable (`Stopwords+SB+noSym`, .62 ±.01 versus `Stopwords+SB+Sym`, .55 ±.02). We found that including sentence boundaries was helpful (`Stopwords+SB+noSym`, .60 ±.01 versus `Stopwords+noSB+noSym`, .62 ±.01, significance p=0.05), but that commas and sentence location information were not useful (`Stopwords+SB+CA+SL+noSym`, .61±.01).

Despite their connections with discourse structure, discourse connectives (`DiscConn+SB+noSym`, .61±.01) failed to outperform stopwords (`Stopwords+SB+noSym`, .62 ±.01). This may be

due to the rarity of discourse connectives in the discussion turns: Turn pairs have an average of 9.0±8.6 (or 6.5±6.3 if *and* is removed from the list) discourse connectives combined, and 118 different discourse connectives are used.

7.7.2 Error Analysis

We examined five pairs each of *true positives* (TP), *false negatives* (FN), *false positives* (FP), and *true negatives* (TN), one set of four from each fold of the best performing system, `pair1grams-+noSW`. Generally, turns from instances classified negative appeared to be shorter in sentence count than instances classified positive (shown by pairs of texts: TN (3.2±2.2 and 3.0±3.4); FN (3.0±2.2 and 2.2±1.1); versus, TP (4.8±4.7 and 4.4±3.6); FP (7.6±10.3 and 5.2±2.8)). Two of the 20 had incorrect gold classification based on misindentation. (See Chapter 5.3.3 for an investigation of misindentation frequency in the Ewdc.)

FP's. One instance is mis-indented. Four of the five FP's appear to require extensive linguistic analysis to properly determine their non-adjacency. For example, one second turn begins, " 'Linking' just distracts from, but does not solve, the main issue", but linking is not discussed in the earlier turn. To solve this, a system may need to determine keywords, match quotations, or summarize the content of the first turn, to determine whether 'linking' is discussed. In another example, the turns can be respectively summarized as, "here is a reference" and "we need to collectively do X." This pair of summaries is never adjacent. Another FP instance cannot be adjacent to any turn, because it states a fact and concludes "This fact seems to contradict the article, doesn't it?" In the final FP instance, both turns express agreement; they start with "Fair enough." and "Right." respectively. This pattern of sequential positive sentiment among adjacency pairs in this dataset is very rare.

FN's. Among FN's, one pair appears nonsensically unrelated and unsolvable, another is misindented, while another requires difficult-even-for-humans coreference resolution. The other two FN's need extensive linguistic analysis. In the first instance, the first turn begins, "In languages with dynamic scoping, this is not the case,[...]," and the other turn replies, "I'll readily admit that I have little experience with dynamic scoping[...]" This may be solvable with centering theoretic approaches (Guinaudeau and Strube, 2013), which probabilistically model the argument position of multiple sequential mentions of an entity such as "dynamic scoping". The second instance consists of a deep disagreement between the two authors, in which they discuss a number of keywords and topic specific terms, disagree with each other, and make conclusions. This instance may need a combination of a centering theoretic approach, stance detection (to recognize permissible sequences such as **Turn1**:"X is a fact"→**Turn2**:"No, X is not a fact"→**Turn3**:"Yes, X is indeed a fact!", and impermissible sequences such as **Turn1**:"X is a

Aspirin words from features		
acid	asa	aspirin
acetylsalicylic	name	generic

Table 7.2: List of "aspirin" unigrams from high information-gain lexical pair features.

	Topics	
	Aspirin	**Wales**
adjacent pairs	16	7
nonadjacent pairs	0	9

Table 7.3: Sample dataset in which the classifier might learn undesirable associations, such as "all Aspirin-topic turn pairs are positive."

fact"→**Turn2**:"No, X *is* a fact"→**Turn3**:"I agree, X is not a fact"), and lexical semantic relation modeling to solve.

7.7.3 Feature Analysis

We examined the top-ranked features from our most accurate system, `pair1grams+noSW` (accuracy = .66±.01), as determined by information gain ranking, in Table 7.2. Of the five lists of features produced during each of the 5 folds of CV, 12 of the top 20 features were in common between all 5 lists, and 11 of these 12 features contained an n-gram referencing "aspirin": (*acid*, *asa* (an abbreviation for acetylsalicylic acid, the generic name for *aspirin*), *aspirin*, *acetylsalicylic*, *name*, *generic*). We explain the likely cause of the topicality in feature importance in Section 7.8, and run a second set of experiments to control topic bias in Section 7.9.

7.8 Topic Bias and Control

In Section 7.7, we showed that lexical pairs are useful for adjacency recognition with random CV fold assignment. However, it is possible that the system's good performance was due not to the lexical pairs, but to information leakage of learning a topic model on instances extracted from a single discussion.

Topic bias is the problem of a machine learner inadvertently learning "hints" from the topics in the texts that would not exist in another experiment addressing the same task. Consider a sample dataset in Table 7.3, which contains 16 adjacent and 0 nonadjacent pairs from an article on *Aspirin*, and 7 adjacent and 9 nonadjacent pairs from an article on *Wales*.

A model trained on this corpus will probably find lexical pair features such as (*?, yes*) and (*why, because*) to be highly predictive. But, lexical pairs containing topic-sensitive words such as *aspirin* and *generic* may also be highly predictive. Such a model is recognizing adjacency

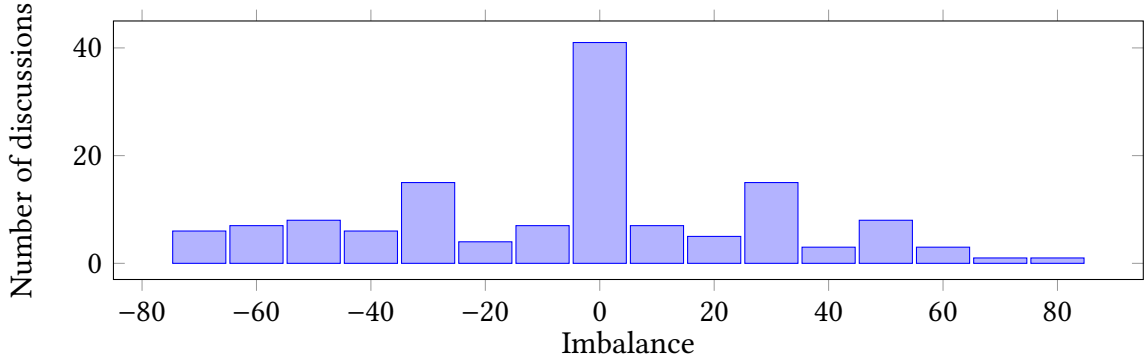

Figure 7.2: Class imbalance by discussion, in percent. -20 means a discussion is 20 percentile points more negative instances than positive; i.e., if there are 10 instances, 4 positive and 6 negative, then the discussion is a -20 discussion.

by topic. To remove this topic bias, instances from a single article should never occur simultaneously in training and evaluation datasets.

Topic bias is a pervasive problem. Mikros and Argiri (2007) have shown that many features besides n-grams are significantly correlated with topic, including sentence and token length, readability measures, and word length distributions. Topic-controlled corpora have been used for authorship identification (Koppel and Schler, 2003), genre detection (Finn and Kushmerick, 2003), and Wikipedia quality flaw prediction (Ferschke et al., 2013).

The class distribution by discussion in our dataset is shown in Figure 7.2; imbalance is shown by the percentage of positive pairs minus the percentage of negative pairs. Only 39 of 550 discussions contributed an approximately equal number of positive and negative instances. 12 discussions contributed only negative instances, and 321 discussions contributed only positive instances[85]. Of discussions with some instances from each class, a whopping 43 of 137 discussions contributed a set of instances that was class imbalanced by 40 percentage points or more. As a result, a classifier will perform above chance if it assumes all instances from one discussion have the same class.

7.9 Experiments with Topic Bias Control

In our second set of experiments, we performed adjacency recognition while controlling for topic bias. To control topic bias, instances from any discussion in a single Wikipedia article are never split across a training and test set. When the cross-validation folds are created, instead of randomly assigning each *instance* to a fold, we assign each *set* of instances from an entire article to a fold. With this technique, any topic bias learned by the classifier will fail to benefit the classifier during the evaluation. We did not use stratified cross-validation,

[85]Many of these discussions may have consisted of only 2 turns.

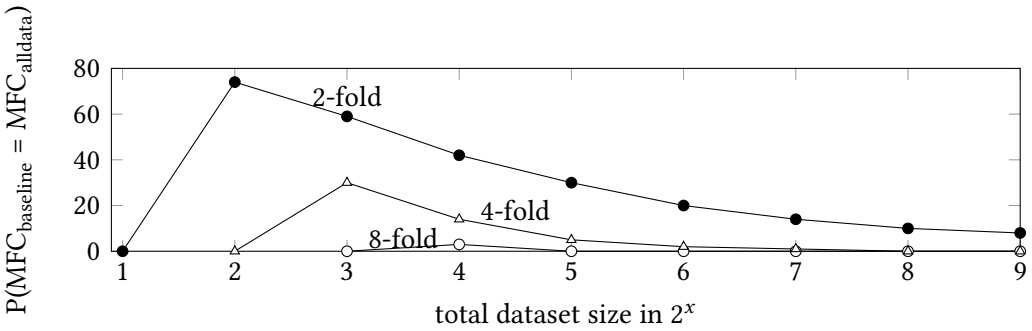

Figure 7.3: Probability of a MFC baseline having the same class-distribution as the overall dataset.

due to the computational complexity of constructing folds of variable-sized threads containing variable class-balance.

7.9.1 Problems with the Chance Baseline

The experiments in Section 7.7 were compared to a *chance* baseline that was very close to the expected classifier performance in the absence of useful features.

However, in our experiments with topic bias control, the expected classifier performance in the absence of useful features is significantly below chance. A classifier faces a statistical bias to classify all instances as the most frequent class of the training set, i.e. the *least frequent class* in the test set. Chance is the upper limit of the MFC baseline, as i instances approaches infinity:

$$\lim_{i \to \infty} f(i) = \text{MostFrequentClass}_{\text{dataset}}$$

When there are few classes and many instances, this does not matter. But, topic bias control effectively reduces the entropy of class distribution in the training and test sets. In Figure 7.2, we showed that many discussions contributed a set of instances with a heavy class bias. This means that, if instances are sorted into folds based on their discussion of origin instead of sorting instances randomly, the entropy of class distribution by fold (and therefore, by training and test set) is reduced. Figure 7.3 shows, for different k in k-fold cross-validation, the probability that an MFC baseline has the same class distribution as the overall dataset. As k increases, and/or the dataset size increases, a MFC baseline becomes less likely to have the same class-distribution as the overall dataset.

To demonstrate the effect of topic bias control on class-imbalance, we calculate class-imbalance and most frequent class baseline on a series of simulated datasets. Figure 7.4 shows that as the number of instances decreases, the probability that the MFC baseline equals 0.5 in a class-balanced binary classification paradigm also decreases. With only 250 instances, there is a significant chance the MFC baseline will be $\leqslant 0.4$.

In our experiments using Topic Bias Control, we compare against the actual MFC baseline, as seen by the classifier in the experiment. The classifier will perform at this baseline if lexical

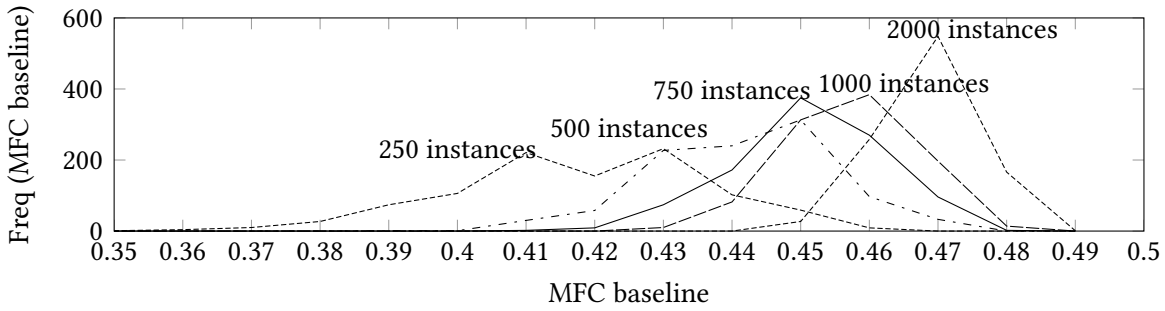

Figure 7.4: MFC baseline distribution for 10-folds and 250, 500, 750, 1000, and 2000 instance datasets.

Feature	Acc w/o TBC	Acc w TBC
MFC	.49±.01	.44±.04
CosineSim	.56±.01	.54±.06
Nonpair1grams	.67±.02	.49±.03
Stopwords+SB+noSym	.62±.01	.51±.01
Stopwords+SB+Sym	.55±.02	.51±.01
Stopwords+noSB+noSym	.60±.01	.53±.02
Stopwords+SB+CA+SL+noSym	.61±.01	.52±.02
DiscConn+SB+noSym	.61±.01	.51±.02
And-as-for	.54±.03	.49±.03
Pair1grams	.64±.01	**.56±.03**
Pair2grams	.57±.03	.52±.03
Pair1ngrams+noDC	.65±.02	**.56±.03**
Pair1ngrams+noSW	**.68±.02**	.52±.03

Table 7.4: Adjacency recognition, without and with topic bias control.

pairs are not useful for the task. We also compare against cosine similarity, similarly to our previous experiments. The *non-pair 1grams* baseline uses an SVM classifier trained over 5000 individual unigrams from the turn pairs.

7.9.2 Results

The results of our topic bias controlled experiments are shown in Table 7.9. As entropy decreases with more folds, to avoid exaggerating the reduced entropy effect, 5-fold cross-validation is used. All other experiment parameters are as in Section 7.7.

All experiment combinations were $p \leqslant 0.05$ significantly different (McNemar, 1947) from the `CosineSim` and `MFC` baselines, except `Stopwords+SB+CA+SL+noSym`, and all were significantly different from the `Nonpair1grams` baseline. Absolute classifier performance in the topic bias control paradigm drops significantly when compared with results from the non-topic-bias-control paradigm. This indicates that the classifier was relying on topic models for adjacency recognition. Not only is the classifier unable to use its learned topic model on the test dataset, but the process of learning topic modeling reduced the learning non-topic-model

feature patterns. Even the feature group `And-as-for` drops, illustrating how topic can also be modeled with stopword distribution, even though the stopwords have no apparent semantic connection to the topic.

The benefit of pair n-grams is shown by the significant divergence of performance of `Non-pair 1grams` and `Pair1grams` in the topic bias control paradigm (.49±.03 versus .56±.03, respectively).

However, several feature sets are still significantly effective for adjacency recognition. `Pair1grams` and `Pair1grams+noDC` perform well above the `MFC` baseline, cosine similarity baseline, and `Non-pair 1grams` baseline. They also outperform the stopword and the discourse connectives feature sets. The shorter n-grams of `Pair1grams` continue to outperform the bigrams in `Pair2grams`, similarly to the experiments without TBC.

Performance of feature sets exceeding the `MFC` baseline indicates that lexical pair features are informative independently of topic bias.

7.10 Chapter Summary

In this chapter, we described the use of lexical pairs as features for pairwise classification of Wikipedia discussion turns for adjacency recognition. Adjacency recognition is an important step in thread reconstruction.

We answered the following questions, posed at the beginning of this chapter:

> *Research Question*: Are lexical pairs of discourse connectives, stopwords, unigrams, or bigrams effective for adjacency recognition? Does adding discourse information or removing stopwords or adding feature symmetry help?

We have shown that the use of lexical pairs is helpful for adjacency recognition, outperforming cosine similarity. We have found that adding discourse information, removing stopwords, and adding feature symmetry are not helpful for adjacency recognition.

> *Research Question*: Is topic bias inflating these results?

We have shown that topic bias was inflating the results of our first experiments. After introducing techniques to counteract topic bias, we showed that the benefit of lexical pairs is robust to topic bias control.

In the next chapter (Chapter 8), we investigate adjacency recognition using a different technique: lexical expansion via human-compiled lexical semantic resources. The adjacency recognition techniques discussed in this chapter plus the next chapter, combined with the thread disentanglement techniques discussed in the previous chapter (Chapter 6), cover the natural language processing component of thread reconstruction.

Chapter 8

Lexical Expansion for Recognizing Adjacency Pairs

Lexical chaining, or the repetition of words and phrases between a discussion turn and its reply, has been found to be useful for adjacency recognition (Aumayr et al., 2011; Balali et al., 2014; Wang et al., 2011a). And previously, in Chapter 6, we have shown how text similarity metrics can be used for thread disentanglement. However, text similarity based on string-match alone will not detect similarity in texts of semantically-related terms. This is especially critical in discussion turn texts: many discussion turns, including emails and Wikipedia discussion turns, are very short, and even turns whose adjacency is obvious to humans may have few or no content words in common.

To draw connections between semantically-related but non-string-matching tokens, we propose to use *lexical expansion*. Lexical expansion, or the expansion of a list of terms with lexical-semantic related terms obtained from a lexical resource such as a dictionary or a distributional thesaurus, has previously been investigated for use in a wide range of NLP tasks such as information retrieval (Voorhees, 1994; Fang, 2008), ad search (Broder et al., 2008), and multi-document summarization (Nastase, 2008; Vanderwende et al., 2007).

In this chapter, we investigate knowledge-rich methods of adjacency recognition. We extract *keyphrases* (terms that best describe a text) from discussion turns, and learn a model of adjacency recognition based on similarity of the sets of keyphrases for pairs of turns. Then we lexically expand a list of *terms* (keyphrases or nouns) from the turn, using handcrafted lexical resources, to create a larger representation of the topic of discussion for each of the discussion turns, and we again learn a model of adjacency recognition based on term similarity. We address the following research questions:

Research Question: Is keyphrase similarity effective for adjacency recognition?

Research Question: Is lexical expansion of terms effective for adjacency recognition?

Research Question: For adjacency recognition via lexical expansion, are keyphrases more effective than nouns as expanded terms?

The chapter is structured as follows. First, we provide an overview of our motivation (Section 8.1), and a discussion of previous research (Section 8.2). Our datasets are presented in Section 8.3. We analyze keyphrases in one of our datasets in Section 8.4. An overview of our experiment design is provided in Section 8.5, and a description of our features in Section 8.6. We discuss our results in Section 8.7. We comment on likely causes of error in Section 8.8.

The material in this chapter has not yet been published.

8.1 Overview

Widespread interest has surrounded the use of handcrafted lexical semantic resources for statistical natural language processing. Resources have been made available, ranging from WordNet (Miller et al., 1990) to UBY (Gurevych et al., 2012). For a variety of NLP tasks, access to semantic relations between words promises unparalleled accuracy. Rahman and Ng (2011) illustrate the benefit of world knowledge for coreference resolution in connecting the noun phrases *Martha Stewart* and *the celebrity*. Voorhees (1994) shows how expanding a search query such as *golf-stroke* to include *golf*, *stroke*, *swing*, *shot*, *slice*, etc., can significantly improve a "less complete query." Barak et al. (2009) propose that categories in a text categorization task can be augmented with words that refer *specifically* (emphasis original) to the category name, such as *pitcher* for the category `Baseball`.

Adjacency recognition is the task of recognizing reply-to relations between discussion pairs. Previous work has shown that knowledge-poor techniques, such as lexical pairs and cosine similarity, can be used to recognize adjacency relations (Jamison and Gurevych, 2014b). However, we propose that the lack of semantic knowledge may hurt system performance. Consider the example in Figure 8.1, from English Wikipedia Discussion Corpus (EWDC) (Ferschke, 2014).

In Figure 8.1, Turn 1 discusses a number of details of Lincoln's health, but does not actually contain the word *health*. Instead, it contains *diseases*, *depression*, *Marfan syndrome*, *disorder*, *multiple endocrine neoplasia type 2B*, *medical conditions*, etc. Turn 2 summarizes the topic with *medical*, *mental health*, *health*, *Abraham Lincoln*, etc. The only topic-specific content words shared between Turn 1 and Turn 2 are *medical* and *Lincoln*. The adjacency of this pair of turns might be better modeled if we could connect *mental health* with *diseases* and *depression*, and *health* with *Marfan syndrome*, *disorder*, *multiple endocrine neoplasia type 2B*, *medical conditions*, etc.

In addition to topic-specific content words such as *diseases*, *multiple endocrine neoplasia type 2B*, *disorder*, etc, the turns contain many other n-grams, such as *on this* and *well known*, and non-topic-specific nouns such as *mention* and *this page*. It would not be useful to investigate lexical semantic connections between such words for adjacency recognition.

Turn 1: *I had a hard time finding any mention of Lincoln's* ***diseases*** *on this page. His well known* ***depression****, or "melancholy," is only mentioned in Marriage and family. And there is no mention at all of the debate as to whether or not he suffered from* ***Marfan syndrome****. There (the Marfan article,) this quote: "Abraham Lincoln may or may not have had Marfan's syndrome, although he undoubtedly had some of the normal characteristic features.[50][51][52] According to a 2007 theory, it is perhaps more likely that he had a different* ***disorder****,* ***multiple endocrine neoplasia type 2B****, that caused skeletal features almost identical to Marfan syndrome.[53]" I remembered I'd seen it on WP, but couldn't even find that through "What links here;" I had to Google it from outside WP. Seems to me there should be a section about his* ***medical conditions****, including the symptoms he suffered from, and the possible diagnoses commonly suggested today. If it's best to put it on a separate page, (this page is really long,) then there should be a leading paragraph here, or at least a wikilink under See also.*
Turn 2: *yeah;* ***Medical*** *and* ***mental health*** *of Abraham Lincoln covers that and it's linked under Marriage and family.*
Adjacency Status: Positive

Figure 8.1: An example adjacency pair that could be better recognized with lexical semantic knowledge. Terms that are semantically related to *medical* and *mental health* are **boldfaced**.

In order to identify topic-related content words that may be connected by lexical semantic relations, we extract *keyphrases*. Keyphrases are a set of terms that best describe a text (Mihalcea and Tarau, 2004), providing a summary of the text that might be helpful for information retrieval (Erbs et al., 2014).

In this chapter, we extract keyphrases from discussion turns, and learn a model of adjacency recognition based on weighted cosine similarity of the sets of keyphrases for pairs of turns. Two turns with a high expanded lexical overlap feature a high similarity score; the scores function as machine learning feature values. Then we lexically expand a list of *terms* (keyphrases or nouns) from the turn, using handcrafted lexical resources, to create a larger representation of the topic of discussion for each of the discussion turns, and we again learn a model of adjacency recognition based on term similarity. We find that keyphrase terms and lexical expansion, while outperforming such naive baselines as majority class, fail to outperform knowledge-poor approaches. This is true regardless of whether all nouns or only keyphrases are used to populate the list of terms. Our error analysis suggests that this is due to poor keyphrase extraction, as well as spurious semantic connections resulting from poor word sense disambiguation during lexical expansion.

8.2 Related Work

Handcrafted lexical semantic resources have been used to expand terms and enhance semantic knowledge in a wide range of NLP tasks.

Information Retrieval Lexical expansion has been extensively investigated for search queries in information retrieval (IR). In Voorhees (1994), an early work on query expansion, search queries are expanded via WordNet one-degree relations of nouns, including `is-a`, `part-of`, etc. Query terms to be expanded are chosen by hand, such that the results reflect an upper-bound on real-world performance. The evaluation shows that, even with hand-picked queries, "relatively complete" queries are not benefited by query expansion, although "less well developed" queries show improvement.

Previous work has shown some limited benefit of search query expansion via WordNet. Fang (2008) proposes a tf-idf weighting alternative based on the overlap of synset definitions. Evaluation on the six TREC collections shows a significant benefit over non-expansion.

Previous work in IR has not consistently shown positive results in the use of query expansion to search for "concepts" instead of specific terms. In Lin and Demner-Fushman (2006), it is hypothesized that this query expansion failure was due to the multi-domain nature of previous evaluations. They find that evaluation on the medical research domain, using domain-specific resources, shows benefits of query expansion over a state-of-the-art domain-generic approach.

Greenberg (2001) investigates query expansion of ABI/Inform business school database by lexical relations from the lexical resource ProQuest Controlled Vocabulary (synonyms, hyponyms, hypernyms, and "related terms") and finds that query expansion via synonyms and hyponyms is effective for IR, with improved recall and insignificantly reduced precision, while query expansion via related terms and hypernyms is not effective, with improved recall but significantly reduced precision.

Ad search Knowledge-rich features have also benefited ad search. In the task of ad search, a few carefully-selected advertisements are displayed next to the standard user query results. The challenge is to select ads that are related to the user's search query. In Broder et al. (2008), ad search queries are expanded by a bag of words from query document results; by using a human-built taxonomy of 6000 types plus their examples as a source of features; and by an auto-extracted distributional list of significant phrases. Evaluation showed that query expansion outperforms both non-expansion and a log-based query substitution system.

Efron et al. (2012) point out that the difficulty in IR of all short texts, including ads, is two-fold. First, there is a vocabulary mismatch problem, a high risk that query terms are not mentioned in relevant short documents. Additionally, tf-idf weighting variants may be ineffective because most terms only occur once in a short document leading to bad topic modeling. Efron et al. (2012) propose a solution to these problems by preprocessing the collection of short documents to augment each document with language model statistics that result from submitting the short document itself as a pseudo-query against an informative corpus, and calculating language model statistics from the top n documents returned from that corpus. An evaluation of this technique on a corpus of Twitter posts showed up to 4-5 p.p. MAP score

improvement, and the Digital Library Metadata Collection showed up to a 9 p.p. MAP score improvement.

Multi-document summarization MDS is a multi-stage task that includes retrieving relevant documents, extracting relevant sentence candidates from these documents, and shortening or combining the sentences to avoid redundancy. Nastase (2008) uses Wikipedia and WordNet as lexical resources for query expansion used at several stages of a multi-document summarization process. First, link text from the first paragraph of the Wikipedia article for each query term, and hypernyms, hyponyms, and antonyms from WordNet for each query term, are used for query expansion in the information retrieval stage of gathering topically-relevant articles for summarization. Later, the expanded query terms are used to select and weight sentences from the topically-relevant retrieved documents, for inclusion in the topic summary. Evaluation on DUC 2007 data showed that query expansion with Wikipedia was slightly helpful, increasing ROUGE score performance by about 1 p.p., while the benefit of WordNet was less clear.

In Vanderwende et al. (2007)'s work on multi-document summarization based on a topic query, a system using an automatically-created thesaurus of synonyms from online clustered news documents is used for lexical expansion of the topical query when weighting sentences as candidates for inclusion in the summary. WordNet was also tried as a lexical resource, but results were clearly negative. In the evaluation on the DUC 2005 and 2006 datasets, it is unclear that query expansion via auto-created thesaurus consistently or significantly improved results, but the system ranked very well compared to other systems in the task.

Question Answering Lexical semantic resources have been used for the task of question answering. Semantic frame resources (FrameNet, PropBank) can also be used to identify lexical semantic relations, through similarities in the frame structure of semantically-related verbs. Narayanan and Harabagiu (2004) use FrameNet and PropBank to abstract over questions in a Question Answering task, resulting in higher coverage of answers than systems that rely on string-matching between questions and candidate answers.

Chu-Carroll et al. (2006) give an overview of a multi-component question answering system. One component, the *knowledge source adapter*, reformulates knowledge from diverse resources such as US Geological Survey and WordNet, so that it is accessible to queries. They find that adding two more knowledge sources, increases system performance 19.9% in relative improvement in percentage of correct answers.

Previous work has shown that bag-of-words surface-form word matching produces poor precision accuracy in question answer sentence selection. Yih et al. (2013) implement several lexical semantic models (polarity-inducing LSA created from the Encarta thesaurus; hypernymy relations from WordNet and Probase; and three vector space models (VSMs) of word similarity) to connect semantically-related but non-string-matching terms between question

and answer sentence candidates. The lexical semantic models result in an improvement over both identical word matching and lemma matching.

Recognizing Textual Entailment Lexical semantic resources have also been used for the task of recognizing textual entailment (RTE). Previous research has shown that syntactic tree kernels are useful for RTE. Mehdad et al. (2010) expand on the concept of tree kernel similarity between the text and hypothesis by matching tree fragments of semantically similar end nodes (words). They find that these syntactic semantic tree kernels (SSTK's) are not more effective than regular syntactic tree kernels (STK's) with WordNet path-based similarity but are more effective with distributional semantic resources.

Mirkin et al. (2009) perform a comparative evaluation of the performance of seven pre-existing lexical relation algorithms and their respective lexical resources (WordNet, Wikipedia, etc) for a textual entailment task. They identify different factors affecting rule applicability and resource performance, noting that in general, useful lexical relationships are embedded among many irrelevant ones.

Coreference Resolution Some approaches to coreference resolution use knowledge-rich features. Rahman and Ng (2011) describe a coreference resolution system which includes semantic features encoding world knowledge from YAGO, a database of facts extracted from Wikipedia and WordNet, and FrameNet. The features are designed around the observation that lexical semantic knowledge indicating non-coreference (*lion*, *tiger*) is equally important to that indicating coreference via WordNet path distance, etc. They find that world-knowledge systems outperform a strong knowledge-poor baseline system.

Ponzetto and Strube (2006) introduce coreference resolution machine learning features using the semantic knowledge sources WordNet and Wikipedia. WordNet features measure path-based and information-content similarity between the referring expression and antecedent candidate, using highest and average scores across all senses of each word. Wikipedia features are extracted from the Wikipedia pages of the referring expression and antecedent candidate, such as whether the first paragraph mentions, links to, or contains the category of the other page, the gloss overlap between the first paragraph of the two pages, and the information content relatedness score of the two pages from the Wikipedia directed graph of the categories. They found that the lexical semantic features improve system recall as well as overall F_1.

Information Extraction Lexical resources have been used for information extraction. Chai (2000) learns automatic information extraction rule generation by mapping tokens from a corpus to WordNet synsets and using the verb or preposition in-between the two nouns to denote the function. WN synsets allow the token to be replaced by synonyms or hypernyms to learn more general rules. The evaluation tunes the parameters to learn which WN replacement de-

grees are helpful. Evaluation was on a corpus of job advertisements for a newsgroup. Synonym use improved F_1 over exact tokens, but hypernyms did not.

Bordes et al. (2012) learn meaning representations via joint modeling of knowledge acquisition and word sense disambiguation, using WordNet to learn along with raw text. A learned ranking function results in improved ranking precision on knowledge acquisition and improved F_1 on WSD with WordNet-based modeling, over a random baseline.

Chan and Roth (2010) perform relation extraction (RE) by predicting, for a predefined set of relations, whether any of the relations exists between a pair of mentions m_1 and m_2 in a sentence. One of the techniques used is to map each mention to a predicted Wikipedia page for the mention, and use the content of the pair of pages to determine 1) relations, if any, and 2) hyponymy. The Wikipedia technique improved upon the baseline RE system by 1 p.p. F_1.

Text Categorization Lexical resources have also been used for text categorization. Barak et al. (2009) describe a text classification system in which lexical relations of the category classes (synonyms, hyponyms, meronyms, etc.) are included in the feature space. Evaluations on 20NewsGroups and Reuters-10 datasets indicate improved performance over noisy context models (i.e., bag of words) alone.

Gabrilovich and Markovitch (2005) collect "most characteristic" n-grams from websites that are linked to or categorized in the knowledge source Open Directory Project (ODP) as associated with the "concepts" of the text being classified. Concepts include original concepts plus "is-a" linked concepts in the knowledge source hierarchy. The terms from the website text of the linked websites, used as features, result in improved text classification on the Reuters-21578, Reuters RCV1, 20 Newsgroups, and movie reviews corpora.

Çelik and Gungor (2013) enhance the feature space in a text classification task with semantically related terms from WordNet, and find that the lexical semantic system slightly outperforms text classification using n-grams alone. They also observe the lexical semantic problem of unrelated word senses, and propose a rough WSD technique to discard features from synsets that have too few terms in common with other synsets of the document.

8.3 Datasets

For our investigation of semantically-informed adjacency recognition, we used two datasets. One dataset consists of pairs of turns from the Enron Threads Corpus (Etc) (Jamison and Gurevych, 2013) and contains 4995 turn pairs. 861 pairs comprise 342 sets with one shared email, and another email from the same thread; there is only one adjacent pair per set. A set of pairs with one shared email is illustrated in Figure 8.2. The additional 4134 singleton pairs are not part of evaluation sets but are added to the training data for better model learning. A sample adjacent email pair is shown in Figure 8.3.

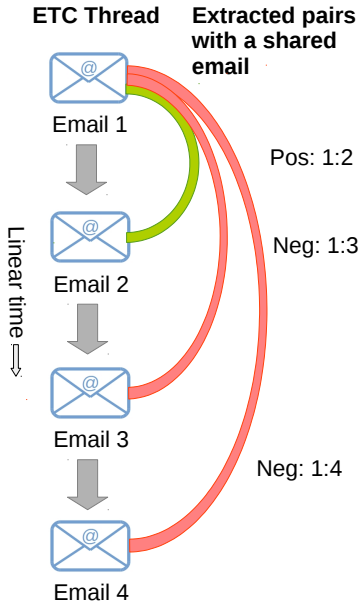

Figure 8.2: An original thread displayed in time-linear order, and a set of email pairs with one shared email.

Turn 1: *Jeff at Grant confirmed that we cannot use the green as small hydro it was already allocated.*

Turn 2: *R: It looks like we cannot use the Grant stuff as green. C*

Adjacency Status: Positive

Figure 8.3: An example adjacency pair from the ETC dataset.

The other dataset consists of 4047 pairs of turns from the English Wikipedia Discussions Corpus (EWDC) (Ferschke, 2014). 1547 pairs comprise 440 sets with one shared turn, and another turn from the same discussion; there is only one adjacent pair per set. 2500 pairs (1500 positive, 1000 negative) that are not part of a set were added to the training data. A sample pair is shown earlier in Figure 8.1.

8.4 Keyphrases in Wikipedia Discussions

We investigated the frequency of semantically-similar keyphrases in adjacency pairs by examining keyphrases from adjacent-classified turns. Keyphrases from a subset of the EWDC are extracted by TextRank (Mihalcea and Tarau, 2004), an unsupervised graph-based ranking model for text processing. TextRank produces state-of-the-art keyphrase extraction in short texts (Erbs, 2015). The TextRank keyphrase extraction provided by DKPro Keyphrases[86], a freely-available keyphrase extraction framework (Erbs et al., 2014). We used state-of-the-art adjacency recognition (Jamison and Gurevych, 2014b), based on lexical pairs, and analyzed the results.

We examined 15 turn pairs each of false negatives (FN), false positives (FP), and true negatives (TN), for the following:

[86]https://github.com/dkpro/dkpro-keyphrases/

	FN	TN	FP
Ideal KP similarity	.73	.33	.47†
Actual KP similarity	.27	.20	.40†
Expand KP similarity	.33	.27	.40‡
SemRel KP pairs	11 prs	7 prs	16 prs

Table 8.1: Analysis of keyphrase similarity in FN and negative classes from an evaluation with current state-of-the-art adjacency recognition. †Two pairs were a corpus classification error. ‡One pair was a corpus classification error.

Ideal KP similarity Percentage of pairs that should have >0 keyphrase (KP) similarity, if ideal keyphrases are extracted. For this task, ideal keyphrases are all nouns that are not overly general (such as *thing*, *facts*, and *mention*) and are not part of a non-compositional phrase (such as "a hard *time*").
Actual KP similarity Percentage of pairs with >0 keyphrase similarity, using TextRank keyphrase extraction.
Expand KP similarity Percentage of pairs with semantically related TextRank keyphrases. These pairs are expected to be affected by lexical expansion.
SemRel KP pairs Total number, among all keyphrases from all 15 turn pairs in this category, of keyphrase pairs (not turn pairs) where semantic expansion would connect two keyphrases.

Our analysis focuses on the false negative (FN) category. We expect that an adjacency recognition feature measuring cosine similarity between sets of lexically-expanded keyphrases will have the effect of increased positive recall and decrease positive precision. The current best (knowledge-poor) adjacency recognition system, which uses lexical pair features, has a sparse feature space and many positive instances have all 0-value features. Our new proposed lexical expansion cosine similarity features will reduce the number of instances with all 0-value features. Therefore, it is particularly interesting to inspect keyphrases from the FN's of our previous-best system. Table 8.1 shows the results of our investigation.

73% of FN's should have >0 keyphrase similarity, and 27% actually do. While there is room for better keyphrase extraction, a feature reflecting keyphrase similarity should increase positive recall for this dataset. However, 20% of TN's and 40% of FP's also have >0 TextRank keyphrase similarity, so decreased positive precision may offset the increased positive recall.

33% of FN's should have >0 semantically-expanded keyphrase similarity, with an average of 2.2 matches per instance. 27% of TN's and 40% of FP's should have >0 lexically-expanded keyphrase similarity, with respective averages of 1.75 and 2.67 matches per instance. While our sample is too small to calculate whether this difference is significant, it is possible that, like keyphrase similarity, the decreased positive precision may offset the increased positive recall.

FN	TN	FP
fact:opinion	American:people	Google:site
MED:Medicine	hurricane:Anita	Google:web
WikiProject Medicine : WP:MED	hurricane:Alicia	ranking:site
fact:statistic	Africa:Africans	war:conflict
culture:ethnicity	Black:Whites	waterfall:tributary
origin:nationality	quote:statement	radio:news
culture:nationality	religion:faith	Falls:tributary

Table 8.2: A sample of semantically-related keyphrase pairs.

Table 8.2 shows a sample of the keyphrase pairs from Table 8.1 that have semantic similarity. As can be seen by the examples, a wide range of semantic relations are found in the turn pairs. Relations include antonyms (fact:opinion), synonyms (fact:statistic), related(radio:news), and more. In our experiments, we will represent these different relations as different machine learning features.

8.5 Experiment Design

In this chapter, we approach adjacency recognition as a ranking task: given a set of potentially-adjacent pairs in which one pair is adjacent and the rest are not, can the system identify the positive pair by scoring it with a greater likelihood of adjacency than the others. Each set of potentially-adjacent pairs shared one turn. Additional non-set pairs were added to the training data to increase training set size. This ranking strategy was chosen in order to model the problem that some discussions might contain much more topic-relevant vocabulary than others, such that a linear classification model would be ineffective at applying a class division threshold learned on one discussion to turns of a different discussion.

Our evaluation measures whether the gold adjacent pair was the top ranking pair of the set. In this ranking classification scenario, a confusion matrix of each pair's classification always yields equal numbers of False Positive (FP) and False Negative (FN) results, which causes Precision, Recall, and F_1 to be equal. Therefore, we report results in *accuracy*:

$$accuracy = \frac{AdjacentPairs \cap TopRankedPairs}{AdjacentPairs}$$

This is fundamentally equivalent to the $accuracy_{edge}$ metric used by Seo et al. (2011) and the accuracy used by Balali et al. (2014) for evaluation of discussion threads, but described differently, since their tasks do not explicitly describe negative turn pairs.

Experiments used 5-fold[87] cross-validation, and an SVM regression learner (Weka's SMOreg) learned a probability of adjacency for each pair. Using the output regression scores for each pair, we calculated ranks separately for evaluation.

8.6 Features

Each machine learning feature is a weighted cosine similarity value, ranging 0.0-1.0, that reflects the similarity of the terms list from Turn 1 and the terms list from Turn 2. A terms list may consist of just original terms (keyphrases or nouns), or original terms plus their lexical expansions, as determined by one or more lexical relations in the lexical resource Uby.

As previously described in Section 8.4, keyphrases are extracted from discourse turns via the TextRank algorithm (Mihalcea and Tarau, 2004) from the DKPro Keyphrases framework (Erbs et al., 2014). Expansion uses only the first (i.e., most frequent) sense of the term; this naive metric is difficult for highly advanced WSD systems to surpass[88]. We selected keyphrases of syntactic types *noun*, *noun chunk*, and *proper noun chunk*. Terms (including expansion terms) are weighted based on the keyphrase values computed by TextRank. An example keyphrase list for the turns shown in Figure 8.1 is shown in Figure 8.4.

Turn 1 Keyphrases: *Marfan, syndrome, page, mention, Lincoln*
Turn 2 Keyphrases: *Medical, Marriage, health, Lincoln, Abraham*

Figure 8.4: Extracted keyphrases from Figure 8.1.

The keyphrases in Figure 8.4 are not necessarily ideal: in Turn 1, *depression* and *medical conditions* would be better keyphrases than *page* and *mention*. Therefore, as an alternative to selecting keyphrases as a subset of the text, we also tried using all nouns as terms.

We used Uby, a large-scale lexical semantic resource framework that combines information from several handcrafted lexical resources, to access lexical relations from WordNet, Wiktionary, Wikipedia, FrameNet, and OmegaWiki. A full list of our lexical relations is shown in Table 8.3. Each relation is used to produce a turn pair's sets of expanded terms, and the cosine similarity of those expanded terms is one feature. Features are combined into the feature groups shown in Table 8.4.

[87] 5-fold was chosen instead of the conventional 10-fold, due to computational complexity of the experiments and Uby database lexical resource.

[88] In Senseval-3, only 5 of 26 systems outperformed the most frequent sense baseline, and none exceeded it by >3 percentage points (Snyder and Palmer, 2004).

		Example	
	Relation	**original term**	**expanded new term**
Names	holonym part	tree	bark, leaves
	hypernym	Audi	car
	hypernym instance	Mississippi river	river
	hyponym	car	Audi
	is topic of	medicine	acute, chronic
	meronym part	car	wheels, seats
	meronym substance	bread	flour
	related	magazine	paper
	subsumed by	indicator	chronometer
	subsumes	name	first name
	topic	conflict	military
Types	association	vocabulary	grammar, word
	complementary	war	peace
	taxonomic	case	dative case
	label	Navy SEAL	armed forces
	part whole	vocabulary	speech
	predicative	deer	fawn

Table 8.3: Lexical semantic relations and examples.

8.7 Results

Experiment results are shown in Table 8.5. Almost all keyphrase and lexical relation feature groups (except *RelNames* with emails) outperform a chance baseline. All keyphrase and lexical relation features combined (*AllLexRel*) have the best performance of any feature group derived from keyphrases, with performance 6 percentage points (pp.) (ETC) and 17 pp. (EWDC) above chance.

However, cosine similarity over all words (*CosineSimAllWords*) is a much stronger feature. Alone, it outperforms all semantic similarity-based feature sets, with performance 10 pp. (ETC) and 26 pp. (EWDC) above the chance baseline. When combined with other feature groups (*AllPlusCSAllWords*), performance is lower than *CosineSimAllWords* alone. Neither the keyphrase features (*CosineSimPlusKeyph*) nor the knowledge-rich features (*AllPlusCSAllWords*) improve the recognition of adjacency pairs beyond *CosineSimAllWords*.

And, lexical pairs of unigrams (*LexPairUnigrams*), a knowledge-poor approach related to n-grams (Jamison and Gurevych, 2014b), outperforms all other feature sets, with performance 13 pp. (ETC) and 39 pp. (EWDC) above the chance baseline.

To test if suboptimal keyphrase extraction was reducing knowledge-rich feature performance, we compared systems with knowledge-rich features that were based on extracted keyphrases to those based on all nouns. The results are shown in Table 8.6. When cosine

Feature group	Summary	# Feats	Expanded Relations
RelNames	Cosine sim of terms with lexical expansion via individual lexical relations	9	hypernym, hypernym-instance, is-topic-of, meronym-part, meronym-substance, related, subsumed-by, subsumes, topic
RelTypes	Cosine sim of terms with lexical expansion via groups of lexical relations in a typology	6	association, complementary, taxonomic, label, part-whole, predicative
Keyphrases	Cosine sim only original TextRank keyphrases	1	(none)
AllLexRel	Cosine sim of terms with lexical expansion via all relation names, relation types, and original keyphrases	19	hypernym, hypernym-instance, is-topic-of, meronym-part, meronym-substance, related, subsumed-by, subsumes, topic, association, complementary, taxonomic, label, part-whole, predicative, *Keyphrases*, *RelNames* combined, *RelTypes* combined, *RelNames* and *RelTypes* combined
AllPlusCSAllWords	*AllLexRel* plus *CosineSimAllWords*	20	(see *AllLexRel*)
CosineSimPlusKeyph	*CosineSimAllWords* plus *Keyphrases*	2	(none)
CosineSimAllWords	Tf-idf weighted cosine similarity between the sets of all tokens in the pair of turns	1	(none)
LexPairUnigrams	Most frequent pairs of unigrams, as described in Ch.7	750	(none)

Table 8.4: Feature groups and descriptions.

Training	Emails Acc	Wiki Acc
baseline chance	.3972	.2844
RelNames	.3947	.4364
RelTypes	.4386	.4205
Keyph	.4152	.4455
AllLexRel	**.4561**	**.4568**
AllPlusCSAllWords	.4152	.5023
CosineSimPlusKeyph	.4327	.5455
CosineSimAllWords	.5000	.5455
LexPairUnigrams	**.5285**	**.6755**

Table 8.5: Emails and Wiki adjacency pair recognition results.

Training	Emails Acc	Wiki Acc
baseline chance	.3972	.2844
AllLexRel, keyphrases	.4561	.4568
AllLexRel, nouns	.3743	.4750
AllPlusCSAllWords, keyphrases	.4152	.5023
AllPlusCSAllWords, nouns	.4006	.4955

Table 8.6: Comparison of keyphrases versus the use of all nouns.

similarity over all tokens (*CosineSimAllWords*) is included (as in *AllPlusCSAllWords*), there is a small reduction in performance with nouns (ETC: .4152 to .4006; EWDC: .5023 to .4955); we presume this is from increased noise in the feature space by lexical expansion of names, off-topic words, etc, which are unnecessary to represent the full text since *CosineSimAllWords* is already included in the feature set. When cosine similarity is not included as a feature, there is an increase in performance with nouns in the EWDC dataset(.4568 to .4750), but not the ETC dataset (.4561 to .3743). In general, the EWDC dataset shows a higher correlation between adjacency and text similarity; the keyphrase-versus-noun difference with *AllLexRel* may be due to a higher percentage of non-lexically-relatable nouns in the emails. For example, in the emails in Figure 8.3, names such as *Jeff*, *R*, and *C*, as well as jargon such as *the green* and *small hydro*, cannot be usefully lexically expended. Lexical expansion of keyphrases in emails may work best because the five select keyphrases per email are more lexically-relatable than other nouns such as names.

8.8 Error Discussion

Although TextRank has been shown to effectively extract keyphrases from short texts (Erbs, 2015), there is much room for keyphrase extraction improvement, as shown in Figure 8.1. This can be seen by the percentage difference between the ideal and actual keyphrase similarity:

while a class difference in keyphrase similarity between positives (.74) and negatives(.40) can be seen with ideal keyphrase extraction, this difference disappears with TextRank keyphrase extraction (.27 and .30, respectively). Figure 8.5 shows an example of a turn and its ineffective TextRank keyphrases.

Turn: *I agree with Noeticsage, "highly selective" is a nefarious hybrid of a peacocked, weasely, word to avoid. Simply state the facts and let the reader make his or her own determination: X% of undergraduate applicants were admitted, Y% of them enrolled, Z% rematriculated for a 2nd year. Selectivity has absolutely nothing to do with quality, so I don't think it's even a worthwhile distinction to attempt to make. See WP:BOOSTER for more.*
Keyphrases: %, hybrid, WP, word, See

Figure 8.5: A turn and its poor keyphrases. Better keyphrases might include: Noeticsage, highly selective, weasely, undergraduate applicants, selectivity, etc.

However, lexical-expansion knowledge rich features require some form of term extraction, to limit the feature vector space. A similarity comparison of all semantically-related words for all tokens in the turns (including stopwords and words unrelated to the topic of discussion) would introduce too much noise for effective adjacency recognition. And from an intuitive perspective, it makes no sense to investigate the semantic relations of words that are known not to have meaningful semantic relations. Thus, we expect that improved keyphrase extraction might assist in better lexical expansion for adjacency recognition.

An additional challenge is word sense disambiguation during lexical expansion. Consider the turn in Figure 8.6. In this turn, *lead* refers to the beginning section of a Wikipedia article. However, the most frequent word sense, which is used to lexically expand the term, refers to the chemical element. Poor word sense disambiguation will cause spurious semantic connections to be found in nonadjacent turn pairs, because the lexical expansion will result in terms that are unrelated to the discussion of the turn.

Turn: *Opinions should not be featured in the lead, and should be moved to the body of the article. Facts that can be tested and proven correct through scientific and historical documents should always be used in favor of common opinions.*
Keyphrase: lead
Lexical expansion via *taxonomic*: heavy metal, element of group IV, chemical element

Figure 8.6: A turn, one extracted keyphrase, and a sample of its taxonomic lexical expansions

8.9 Chapter Summary

In this chapter, we described the use of lexical expansion via human-created lexical resources for pairwise classification of emails and Wikipedia discussion turns for adjacency recognition. Adjacency recognition is an important step in thread reconstruction.

We answered the following questions, posed at the beginning of this chapter:

> *Research Question*: Is lexical expansion of terms effective for adjacency recognition?

We found that, despite the intuitive appeal of lexical expansion of terms to represent topicality of a text, lexical expansion fails to outperform simple knowledge-poor approaches such as tf-idf cosine similarity and lexical pairs.

> *Research Question*: Is keyphrase similarity effective for adjacency recognition?

Similarly to lexical expansion, and despite similar intuitive appeal of keyphrases to represent topicality of a text, keyphrases fails to outperform simple knowledge-poor approaches such as tf-idf cosine similarity and lexical pairs.

> *Research Question*: For adjacency recognition via lexical expansion, are keyphrases more effective than nouns as expanded terms?

We found that the choice of keyphrases versus nouns as terms for lexical expansion pivots on which machine learning features are used and whether or not many of the nouns in the term are usefully lexically-expandable. Nouns introduce noise into the lexical expansion features, by triggering expansion of more unimportant terms. Conversely, nouns act as a more complete n-gram list, and because n-gram cosine similarity outperforms lexical expansion features, then use of nouns as terms for lexical expansion increase system performance when n-gram cosine similarity is *not* a feature. Finally, when a turn has many nouns that cannot be usefully lexically expanded, such as names and jargon, keyphrases may be more effective.

This chapter concludes our sequence of natural language processing steps for thread reconstruction. In Chapter 6, we have investigated thread disentanglement as a pairwise classification problem. We proposed an automatic thread disentanglement solution using text similarity metrics of the emails. In Chapter 7, we have investigated adjacency recognition as a pairwise classification problem. We proposed an automatic adjacency recognition solution using lexical pairs, a statistical approach to adjacency recognition. In this chapter (Chapter 8), we proposed a knowledge-rich approach to automatic adjacency recognition, using lexical expansion. This concludes our natural language processing contributions for the sequence of thread reconstruction subtasks.

Chapter 9

Conclusion and Future Work

Online discussion is ubiquitous and an increasingly critical component of modern life. From emails to forum discussions to instant message chats to news website commentary, an ever-growing amount of our communication occurs through online discussion.

And yet, when the metadata organizing a discussion is lost, the discussion becomes incomprehensible.

In this work, we have presented a series of NLP-based studies towards the development of a system to reconstruct a discussion thread based on the content of the messages. In our work on crowdsourcing, we studied techniques necessary to produce a new corpus for thread reconstruction as an understudied task: we investigated techniques to reduce the cost of annotation of heavily class-imbalanced data, and we compared techniques for learning the best model from a dataset with redundant crowdsource annotation labels. In our work on thread reconstruction, we divided the reconstruction task into two NLP-based subtasks: thread disentanglement and adjacency recognition. We investigated text similarity measures for use in thread disentanglement. Then, we investigated lexical pairs, as well as knowledge-rich features, for adjacency recognition.

This chapter summarizes the main contributions of this thesis. In Section 9.1, we summarize the general contributions of each chapter. In Section 9.2, we discuss future applications for thread reconstruction technology. To conclude, in Section 9.3, we discuss open issues and limitations of our research.

9.1 Summary of Main Contributions and Findings

We presented **crowdsourcing as a source of NLP annotations** in Chapter 2. We discussed the process and various forms of crowdsourcing and a brief history of crowdsourcing. We discussed demographics of crowdsource workers, and we discussed what annotations tasks have been successful with crowdsourcing. We discussed economic issues of the crowdsource labor

market. Finally, we discuss problems with crowdsourcing label quality, including spam/worker fraud, worker mistakes (accidental/random), worker quality (systemic), worker bias, and the data-driven problem of ambiguity.

Through this overview, we learned that crowdsourcing is a historically-effective labor tool that enables cost-effective linguistic annotation. We saw that different forms of crowdsourcing can provide annotations for a wide variety of tasks. We observed that modern crowdsource work is performed at-will and by a wider demographic than traditional university studies. We reviewed studies showing that crowdsource annotation is reliable in comparison with expensive, trained workers. We also discussed quality-improvement solutions for several known categories of crowdsource annotation problems.

In Chapter 3, we discussed our approaches to **class-imbalanced annotation**, and how annotation costs may be reduced. Because the final desired discussion structure is a directed graph, where nodes are discussion turns and edges are the reply-to relations between the turns, it is necessary that the edges in the discussions corpus are labeled with gold-standard relations. This means that, in practical terms, a human annotator must read many pairs of discussion turns, and label most of the turns as *negative* (i.e., not reply-to) and label a few of the turns as *positive* (i.e., reply-to). Such a class-imbalanced dataset is expensive to annotate, because so much unknown data must be labeled in order to find a few positive instances. By developing techniques to reduce the cost of class-imbalanced annotation, we can enable the production of discussions corpora for later study of thread reconstruction.

We found that a class-imbalanced dataset should not be redundantly annotated, because redundant annotations are very costly in a class-imbalanced annotation task. Specifically, we found that an instance that has received a single common-class label should be presumed to be common-class, and should be discarded during the search for rare-class instances. We showed this to be the case for all three of our class-imbalanced datasets. We also found that, although our rule-based technique of **discarding instances that receive a single common-class label** is radically simpler than our previously proposed technique of identifying rare-class instances using a supervised machine classifier cascade trained on instance metadata, both techniques have roughly the same cost, which is about 70% cheaper than the baseline 5-vote majority vote aggregation. Furthermore, the rule cascade requires no training data, making it suitable for seed dataset production.

In Chapter 4, we bridged the gap from a newly-created crowdsource-annotated corpus, to automatic text classification for thread reconstruction, by investigating how to learn the best machine classifier model from a set of crowdsourced labels. Crowdsource labels are noisy, and it was not clear how to best train a machine classifier on these noisy labels. To provide a generalized comparison between different techniques for **learning a model over crowdsourced labels**, we investigated five different natural language tasks. For each task, we examined the impact of passing item agreement on to the task classifier, by means of soft labeling, and of changing the training dataset via low-agreement filtering.

We found that, in four out of the five natural language tasks, there was a statistically significant **benefit from filtering**, compared to integrated labels. Filtering, the best-performing strategy, showed strongest improvements on Hard Cases. The classifiers were not able to learn from the disagreement of the annotators, and this showed most clearly for borderline instances, the Hard Cases. However, we also observed our training strategies impacted some classification categories more than others, increasing sample selection bias, which negatively impacts model learning. Our findings suggested that the best crowdsource label training strategy is to remove low item agreement instances, although care must be taken to reduce sample-selection bias.

In Chapter 5, we provided the theoretical foundations of **discussion thread reconstruction**. We defined concepts of thread reconstruction and provided examples of online discussion threads and related problems. We discussed previous research that is related to discussion threads and thread reconstruction. Then we explained the **construction of our Enron Threads Corpus** (ETC), and we described the English Wikipedia Discussions Corpus (EWDC) (Ferschke, 2014).

In this overview, we saw that online discussions are pervasive in a wide range of everyday applications, from emails to internet relay chat to Wikipedia discussion pages, to social voting sites like Reddit, to news article comment sections, to question-answering websites. We learned about software advances for thread visual display. We reviewed work modeling the discussion, including thread summarization, as well as work modeling the user, and modeling the post. We reviewed work investigating the effect of design-, automatic-, or moderator-imposed constraints on online discussions. We saw that the discussion or aspects thereof may have an impact in downstream tasks. We also examined work specifically on thread reconstruction, and saw that most such previous work relies on turn metadata. Finally, the descriptions of the ETC and EWDC pave the way for our thread reconstruction research in later chapters.

In Chapter 6, we investigated email thread disentanglement, treating **thread disentanglement as a pairwise text similarity classification problem**. We provided a description of the text similarity features, along with examples motivating their use. We described our disentanglement experiments, firstly in an evaluation with random negative instances and secondly in an evaluation that controls for the semantic similarity of the corpus.

We found that content-type text similarity features are more effective than style or structural text similarity features for pairwise classification email thread disentanglement, and we found that semantic features are ineffective, perhaps due to the domain-specific nature of e-mails. There appear to be more stylistic features uncaptured by our similarity metrics, which humans access for performing the same task. We also showed that semantic differences between corpora will impact the general effectiveness of text similarity features, but that content features remain effective.

In Chapter 7, we investigated the **recognition of adjacency pairs**, as applied to Wikipedia discussion turns. We approached this task as a pair classification task, and we proposed features that are particularly suited for the pair classification paradigm. We provided background information on adjacency pair typologies, and a discussion of previous research. We described our human performance annotation experiment, which was used to determine an upper bound for this task on our dataset. We described and motivated our feature sets (Section 7.6). Our first set of automatic adjacency recognition experiments used no topic bias control. Then we discussed the **problem of topic bias** and solutions for its control. We re-ran our experiments using topic bias control, and compared the results with our non-topic-bias-controlled experiments.

We showed that the use of lexical pairs is helpful for adjacency recognition, outperforming cosine similarity. We found that adding discourse information, removing stopwords, and adding feature symmetry are not helpful for adjacency recognition. Additionally, we showed that topic bias was inflating the results of our first experiments. After introducing techniques to counteract topic bias, we demonstrated that the benefit of lexical pairs is robust to topic bias control.

In Chapter 8, we used **lexical expansion for automatic adjacency recognition**. Lexical expansion is the expansion of a list of terms with lexical-semantic related terms obtained from a lexical resource such as a dictionary or a distributional thesaurus. We applied lexical expansion to an extracted list of terms (keywords or nouns), and used similarity of terms to predict turn adjacency via a series of pairwise ranking experiments.

We found that the choice of keyphrases versus nouns as terms for lexical expansion pivoted on which machine learning features were used and whether or not many of the nouns in the term were lexically-expandable. Nouns introduced noise into the lexical expansion features, by triggering expansion of more unimportant terms. Conversely, nouns acted as a more complete n-gram list, and because n-gram cosine similarity outperformed lexical expansion features, the use of nouns as terms for lexical expansion increased system performance when n-gram cosine similarity was *not* a feature. Finally, when a turn had many nouns that could not be lexically expanded, such as names and jargon, keyphrases might have been more effective.

9.2 Applications

Discussion thread reconstruction technology has the potential to be useful in a variety of applications. In this section, we discuss uses for downstream tasks including: evidence collection (law enforcement); thread manipulation detection (law enforcement); thread organization and display (email client); real-time user-suggestion/correction (email client, forum software); better email client ad search via better word sense disambiguation (email client), discourse generation (chatbots), and finally, post-hoc thread structure correction of user errors (forums, website comments, and Wiki discussions). Thread reconstruction is also the broad super-task

that covers question answering and answer ranking, but because these are stand-alone subdisciplines, we do not discuss them here.

9.2.1 Law Enforcement Applications

Evidence collection In October 2001, the Securities and Exchange Commission (SEC) began the first of several federal investigations into accounting fraud and related charges in the Enron Corporation, an American energy company located in Texas. Among other problems, Enron was manipulating its financial reports for investors by selling its debt and financial risks to *Special Purpose Entities* (Healy and Palepu, 2003), shell companies that Enron controlled but was not legally required to explain in reports. Enron also adopted the practice of *mark-to-market* accounting, in which the estimated current value of the future profit in a signed long-term contract is reported as current income; such future profitability can be very difficult to estimate accurately, resulting in false or misleading reports for investors.

As the SEC continued its investigation, *Arthur Andersen*, a US firm providing accounting, audit, and consulting services to large companies including Enron, shredded several tons of its Enron records (Healy and Palepu, 2003). The shredding destroyed evidence that would have been used by the federal investigations. However, Enron email remained largely intact. Federal investigators seized the email servers, and the roughly 500,000 emails were searched for evidence to aid the investigations.

The Enron investigation was the first investigation to present a massive collection of emails (or any online discussions) as evidence in a trial. Investigators were faced with problems they had never dealt with before. How to process so many emails? There were too many for humans to read and comprehend. Keyword searching? It's doubtful that Enron executives included phrases such as "hide debt", "special purpose entity fraud", and "misleading future profit estimation" in all emails on the topics, if they used them at all. Furthermore, discussions about company actions on these topics probably involved multiple emails in an email thread, so that keywords and concepts were spread out across the thread, and each individual email's meaning was unknown without reading it as part of its entire thread. However, Enron emails at that time did not contain inherent thread identification, so investigators did not have access to some threads.

The investigations against Enron would have been helped by NLP software that could reconstruct the email threads and process the contents so that the thread itself, and not just its individual emails, was searchable by investigators. This would have led investigators directly to the most useful and promising email evidence, for better and more efficient investigation.

Facing a similar problem, Wu and Oard (2005) describe an information retrieval (IR) task where it is important to have email thread structure. They propose the use of archived mailing lists in technical organizations to recover evidence of design rationale when making future changes to standards or products. They provide the sample question, "When was it agreed

that the HTML 3.0 standard would require further revision?" and point out that the relevant email may contain only the body text, "OK - lets do it!" It is hard for humans, let alone an IR system, to determine whether this message is relevant.

To the best of our knowledge, such email thread reconstruction software does not exist. This is a potential future use for the techniques investigated by the thread reconstruction experiments in this thesis.

Thread manipulation detection In November 2012, the director of the CIA, former U.S. 4-star General Petraeus resigned his position over a scandal involving email cover-up. General Petraeus had been having an affair with his official biographer Paula Broadwell, and during the affair, the two communicated by opening anonymous webmail accounts, and saving messages in the drafts folder. These drafts had little or no metadata: no sender, no receiver, no reply-to headers. Without metadata, it would be hard for a third party observer to understand the messages. At the time of publicity, the press commented that this technique of disguise was "known to terrorists and teen-agers alike" (Fisher, 2012).

The affair came to light during a separate cyberstalking investigation. Affairs in the US military are illegal, presumably because the participants become blackmailable and this is a security risk. However, in General Petraeus's case, the investigation also found classified documents on Broadwell's computer, that she should not have had access to. The theory is that General Petraeus provided her with these classified documents to assist her in writing his personal biography. Although it was not proven that Petraeus was the provider of the classified documents, Petraeus resigned and cited the affair as the motivation.

It would have been very easy for Enron executives to employ a similar headerless email thread strategy to disguise incriminating emails from investigators, although it is unknown if this actually happened. Discussion participants can disguise emails in a variety of ways: multiple users can use the same email address (this was originally suspected in the Petraeus investigation), an email address can be opened for a single purpose, one user can create multiple email addresses. Additionally, users can change the `Subject` header and remove previous quoted material inside the email. In this worst-case scenario, a government agency might end up with a bag of emails and no headers or marked thread structure from which to reconstruct the threads and understand the conversation. It would be useful if investigators had the technology and software to reconstruct these disguised threads.

Such email thread reconstruction software is not known to exist. This is a potential future use for the techniques investigated by the thread reconstruction experiments in this thesis.

9.2.2 Email Client Applications

Email organization and display An email client is software that serves to send and receive emails over a network, as well as providing a visually useful display of received emails and a text editor for composing new emails.

An email client can display received emails using different strategies. Some clients display emails listed in order of receipt; regardless of thread or reply, emails are ordered in a single list by their timestamp. Other email clients group emails by thread, and display threads ordered by the thread's most recent activity timestamp. Clients seem to use a variety of `Subject` and `reply-to` header rules to group emails by thread. For example, gmail strips the `Subject` prefix ("AW:") from the email when determining thread membership, while Microsoft Outlook does not.

From a user perspective, the method used to display received emails impacts usability. When emails from one thread are grouped together, it is easy to re-read the rest of the thread, to understand the most recent received email. The displayed emails can also help a user who wants to send an email but has forgotten the intended recipient's address: the user can simply reply to a previous email that the recipient has sent.

Email usage becomes more difficult when thread display breaks down. For example, in a recent group email thread (Agirre, 2015), a committee was discussing arrangements for an upcoming professional conference. Each user would `reply-all` to the entire group, but the discussion topic split down into two separate issues that the committee needed to handle. The email clients did not realize that the original thread had forked into two threads, so it was unclear to the users which topic a sender was discussing, because all the emails were grouped as one thread, with the same `Subject` and participants.

Nearly all email users have experienced emails that were sent to the intended recipient, but were unrelated and not a reply to the email in which they were labeled a reply. A single "thread" that is composed of multiple threads in this manner, is more difficult to understand and search through in the future.

It would be helpful if email clients were equipped with better technology to recognize thread structure (or lack thereof). Such a client would not group together emails on the two topics as discussed by the conference committee, nor would it group together emails where the user was misusing the *reply-to* function to obtain the recipient's email address. This is a potential future application for the techniques investigated by the thread reconstruction experiments in this thesis.

9.2.3 Email ad targeting

Some webmail applications (such as gmail) earn advertising money by displaying topically-relevant ads next to the email. A similar task, ad search based on search query, returns ads based on a search query that the user has entered via search engine. Broder et al. (2008) explain

[Beginning of conversation]
User: *Are apples green.*
Cleverbot: `Leave!`
User: *Why?*
Cleverbot: `I don't like you.`
User: *But aren't apples good?*
Cleverbot: `No.`
User: *What do you like?*
Cleverbot: `Bunnies.`
User: *But not apples?*
Cleverbot: `Apples are okay.`
User: *You just said apples are not good.*
Cleverbot: `No I never say that.`
User: *You said that 4 turns ago.*
Cleverbot: `I just started talking to you.`

Figure 9.1: A sample discussion between Cleverbot and the author of this thesis.

that ad search is difficult because the user-defined search query is short. A similar problem occurs with email ad targeting. In this situation, the webmail app searches for matching ads based on the content of an email message, and displays the retrieved ads on the side of the screen while the email is displayed. If the email is short, it may be difficult to effectively match ads to the email. For example, a keyword from the email might be matched to an ad using a different word sense than was intended in the email. It would be helpful if thread structure information was available to help the email client target ads more accurately from emails. This is a potential use of the results of our thread reconstruction experiments.

9.2.4 Discourse Generation Applications

A chatterbot (such as Cleverbot[89]) is software that generates real-time short messages to converse with a human in simulated intelligence. A frequent goal of a chatterbot is to participate in such life-like conversation with a human that the human believes the chatterbot to also be human. This is known as passing the Turing test. Other goals might be automated customer assistance (Kuligowska, 2015) or knowledge acquisition (Schumaker et al., 2006).

Chatterbots use various algorithms to generate their replies, but many reply on only the previous message. For example, ELIZA, the earliest chatterbot, identified keywords in the preceding comment and inserted them into a template to create a reply (Deryugina, 2010). ALICE, a type of chatterbot with high modern Turing test performance, uses canned comment-reply pairs, which limits its ability to benefit from the entire previous discussion thread (Schumaker

[89]http://www.cleverbot.com/

et al., 2006). An example of a discussion with a chatterbot (Cleverbot) is shown in Figure 9.1. In this example, Cleverbot states that apples are not good ("User: *But aren't apples good?*" "Cleverbot: `No.`") and just a few lines later recants ("Cleverbot: `Apples are okay.`" "User: *You just said apples are not good.*" "Cleverbot: `No I never say that.`"). It is possible that Cleverbot would generate more realistic comments if it could analyze the entire previous conversation with thread structure to generate a reply. This is a potential future use for better understanding of thread structure and discourse parsing. Additionally, and more immediately, our adjacency recognition techniques (as discussed in Chapters 7 and 8) could be used by a chatterbot to rank potential generated chatterbot replies, before the chatterbot prints them.

9.2.5 Real-Time and Post-hoc Thread Structure Correction Applications

One of the most direct potential applications for the thread reconstruction technology discussed in this thesis is thread structure correction. Section 5.1.1 identifies a number of types of online discussion which may suffer from thread structure errors, such as Wikipedia revision discussions, news article comments, and email threads. In all of these discussions, a user selects a comment to reply to, and the user may select a wrong/unintended comment. As the user is composing their comment, a real-time thread structure system could identify that, based on the mismatch between comment and reply, an error is probably occurring, and based on thread reconstruction between the reply and alternative previous comments, identify the intended previous comment. The error and intended previous comment could be presented as options to the user before they have submitted their reply.

Some types of thread structure errors do not become evident until several turns later. For example, the reply "it's fine with me" can be a reply to a number of previous comments. In ambiguous situations such as this, a thread structure detection system might not recognize a thread structure error for several turns, such as when an email user clicks the wrong email to the right recipient to reply to. In this case, it would be helpful to have a thread structure error detection system that attempts to "clean up" threads at a later point in time, and display the fixed threads along with other threads in the application.

9.3 Open Issues and Limitations

In this section, we will discuss remaining general issues and directions for future work.

Crowdsource Annotation Cost Reduction Our results from Chapter 2 showed that the separation of label identification and label confirmation during crowdsource annotation tasks reduces the costs of class-imbalanced crowdsource annotation. But many pairwise class-imbalanced annotation tasks are still too expensive to be feasible. Since we have removed

redundancy to reduce expenses, the remaining options for cost-cutting focus on the annotation task and the workers themselves. Can a better pairwise task interface be developed, and, can we identify and target the best possible workers for the job?

For example, Parent and Eskenazi (2010) cluster dictionary definitions by crowdsource annotation. They suggest breaking the task down into multiple subtasks. Initially, workers read a list of definitions and label the number of general meanings in the list. Using this number as the number of clusters, workers worked on one of two tasks. In the first task, the *global view task*, workers dragged and dropped each definition from the list into the proper "sense box," such that all senses in a box were one cluster. In the alternative task, the *local view task*, workers were only shown a single pair of definitions and had to decide if the pair were related to the same meaning or to different meanings; all possible pairs of definitions were compared in this manner. Afterward, the pairs were assembled into a graph of clusters based on link inter-annotator agreement. Parent and Eskenazi showed that the two tasks had similar and high inter-annotator agreement, but that the local view task had a complexity of $O(n^2)$ which resulted in a cost more than 60 times the global view task, on a theoretical dataset of 30,000 words. Innovative task design is one future avenue to cutting annotation costs for not just class-imbalanced annotation tasks, but all expensive annotation tasks.

Another technique for cutting annotation costs is worker-targeting. The author of this thesis once worked to produce an NLP crowdsource annotation task requiring highly specialized knowledge. While many HITs ask workers to label pictures or identify a company name in a document, this particular task required workers to identify mentions of gene names in medical abstracts, and specifically to differentiate them from names of proteins which might otherwise be string-identical. The task required deep knowledge of biology.

Although a couple Turkers had the necessary background and were able to do the work, the general sparsity of workers prevented general completion of the project. In this situation, a good solution would have been a worker-targeting plan. How could we get the attention of more workers with the necessary background? For example, we could have publicized this task to local university biology students, who might have particularly enjoyed the work and enjoyed utilizing their new education. Such careful matching between task and worker might help reduce crowdsource annotation costs for class-imbalanced tasks, as well as all other expensive forms of crowdsource annotation.

Machine Learning on Redundant Annotation We investigated techniques to better train a machine classifier by informing the machine learner of linguistic ambiguity, as modeled by inter-annotator agreement on an instance. However, several other forms of worker error, such as spam/fraud, worker mistakes, worker quality, and worker bias, are not modeled in our techniques. Two fundamental next research questions are, *How common are each of these problems?* and if any/all of these problems are found to be common, *Can our previous techniques*

(Jamison and Gurevych, 2015), when combined with solutions for these problems, result in a better strategy for model learning?

For example, item-response algorithms (Dawid and Skene, 1979a) iterate over redundant labels and simultaneously derive a gold standard from the labels while estimating worker bias. Such bias-corrected labels could still be used as soft labels in a classifier such as SVM to train a model: the item-response algorithm would address worker bias, while the soft labels would address instance linguistic ambiguity.

Thread Disentanglement We investigated the use of text similarity metrics for learning a email thread disentanglement classifier. However, along with other approaches that model topicality among discussions, this method is vulnerable to the unique topic distribution of each corpus. While the Etc corpus contains email threads on a wide variety of professional and personal topics, the W3C mailing list corpus, created by crawling the w3.org sites' mailing lists, is much more domain-specific in topics; the more threads in the corpus that discuss the same or similar topics, the harder it is to distinguish between their messages by topic.

One technique for thread disambiguation that would not rely on corpus topic distribution is identifying authorship of the messages and using this participant information to assist in constructing the discussion thread graph. For example, consider the email thread graph in Figure 9.2. Each edge between a person represents an email, and the associated probability represents the chance that that email belongs to this thread. Without participant identification and based on text similarity alone, email ranking might be different than if participant information is available.

When participant identification is available, social network analysis (Scott, 2012) can contribute additional information for a discussion model. For example, the network model may be different if the discussion concerns a corrupt versus non-corrupt project within an organization (Aven, 2012).

Adjacency Recognition We presented the results of several investigations of techniques for pairwise adjacency recognition in this thesis, including a variety of lexical pairs (unigrams, bigrams, strings of discourse connectives, etc.) and knowledge-rich lexical semantic similarity. Yet our results clearly show room for improvement. What else makes discussion turns cohesive?

Schegloff (1990) proposes that the *sequence structure* (i.e., turn order) of a discussion and the topic structure of a discussion are analytically distinct and should be modeled independently. Our previous work has discarded this observation, presuming each discussion turn reflected one topic which was very similar to the topic of an adjacent turn.

Yet, there is a deeper form of organization: as Schegloff and Sacks (1973) point out, there is one omnipresent question for all parties to a conversation, "Why that now?" Each contribution to a discussion must be "demonstrably relevant" for the participants, or the discussion

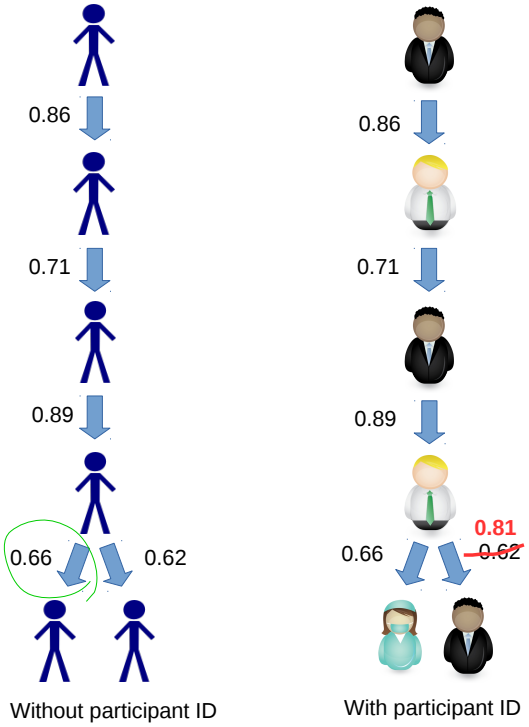

Figure 9.2: Thread reconstruction of a single thread without and with participant identification.

must show evidence of trouble or its suppression. In other words, topical similarity between discussion turns is not sufficient to model coherence.

It's not necessary, either. Consider this discussion from the social voting site Reddit in Figure 9.3. Comment #2 has no topicality in common with Comment #1, yet the discussion is fully cohesive: Comment #2 has switched the discussion to issues concerning the discussion itself, which is shared and relevant world knowledge for all the participants in this conversation. Additionally, Comment #4 is so topically generic that it could be inserted into any discussion at any time; yet, most readers would agree it is a particularly relevant contribution to this discussion in its current position.

In order to account for the observations stated above, and in addition to clause- or phrase-level topic modeling, it would be necessary to account for each participant's world knowledge, relevance between statements or facts, and the information structure that drives order and hierarchy in the search for knowledge. These are incredibly difficult barriers for any computer to model.

9.4 Concluding Remarks

This thesis is a step towards a research pipeline for discussion thread reconstruction, starting from corpus annotation, through model learning, through thread disentanglement, to adja-

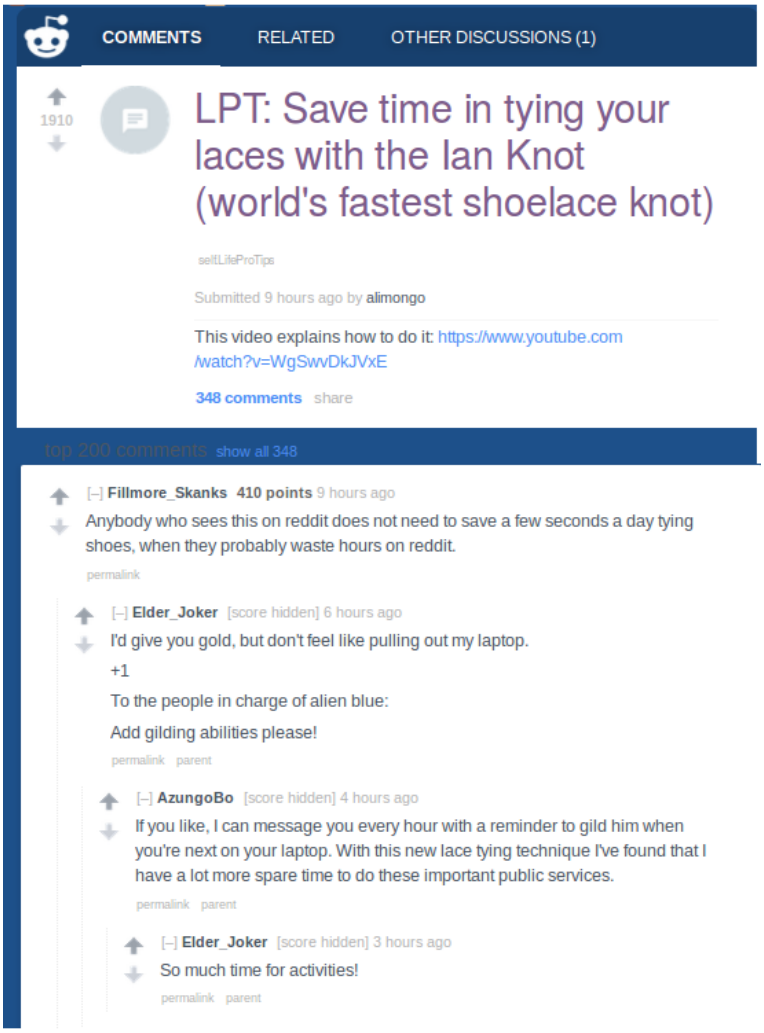

Figure 9.3: Reddit conversation that is coherent without topic similarity between discussion posts.

cency recognition. Although each task was investigated for its contribution towards thread reconstruction, the conclusions from each task contribute towards a far greater range of other NLP challenges, from crowdsource annotation cost reduction for any class-imbalanced dataset, to model learning on any redundantly-labeled corpus, to using text similarity to model conversation topics, to using lexical pairs and lexical semantic knowledge to model turn adjacency in discussion.

List of Tables

List of Figures

Bibliography

Scientific Literature

Héctor Martínez Alonso, Barbara Plank, Arne Skjærholt, and Anders Søgaard: 'Learning to parse with IAA-weighted loss', in: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1357–1361, Denver, Colorado, 2015.

Eneko Agirre: Personal communication, February 2015.

Fernando Alfonso III: 'How Unidan went from being Reddit's most beloved user to its most disgraced', `http://www.dailydot.com/news/reddit-unidan-shadownban-vote-manipulation-ben-eisenkop/`, July 2014. Accessed: 2015-07-31.

Lloyd Allison and Trevor I Dix: 'A bit-string longest-common-subsequence algorithm', *Information Processing Letters* 23 (5): 305–310, 1986.

Vinicius Almendra and Daniel Schwabe: *Fraud detection by human agents: A pilot study*, Springer, 2009.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo: 'A comparison of extrinsic clustering evaluation metrics based on formal constraints', *Information Retrieval* 12 (4): 461–486, 2009.

Paul M. Aoki, Matthew Romaine, Margaret H. Szymanski, James D. Thornton, Daniel Wilson, and Allison Woodruff: 'The Mad Hatter's Cocktail Party: A Social Mobile Audio Space Supporting Multiple Simultaneous Conversations', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 425–432, Ft. Lauderdale, Florida, 2003.

Erik Aumayr, Jeffrey Chan, and Conor Hayes: 'Reconstruction of Threaded Conversations in Online Discussion Forums.', in: *Proceedings of the International Conference on Weblogs and Social Media*, pp. 26–33, Barcelona, Spain, 2011.

Brandy Aven: 'The Effects of Corruption on Organizational Networks and Individual Behavior', *Technical report*, Tepper School of Business, Carnegie Mellon University, 2012, Online: `https://student-3k.tepper.cmu.edu/gsiadoc/WP/2012-E39.pdf`. Electronic proceedings.

A. Balali, H. Faili, and M. Asadpour: 'A Supervised Approach to Predict the Hierarchical Structure of Conversation Threads for Comments', *The Scientific World Journal* 2014: 1–23, 2014.

Daniel Bär, Torsten Zesch, and Iryna Gurevych: 'A Reflective View on Text Similarity', in: *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pp. 515–520, Hissar, Bulgaria, 2011.

Daniel Bär, Torsten Zesch, and Iryna Gurevych: 'DKPro Similarity: An Open Source Framework for Text Similarity', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 121–126, Association for Computational Linguistics, Sofia, Bulgaria, August 2013, Online: `http://www.aclweb.org/anthology/P13-4021`.

Libby Barak, Ido Dagan, and Eyal Shnarch: 'Text Categorization from Category Name via Lexical Reference', in: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 33–36, Boulder, Colorado, 2009.

Gustavo E.A.P.A. Batista, Ronaldo C. Prati, and Maria Carolina Monard: 'A study of the behavior of several methods for balancing machine learning training data', *ACM SIGKDD Explorations Newsletter* 6 (1): 20–29, 2004.

Eyal Beigman and Beata Beigman Klebanov: 'Learning with annotation noise', in: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, pp. 280–287, Singapore, 2009.

Kelly Bergstrom: "'Don't feed the troll": Shutting down debate about community expectations on Reddit.com', *First Monday* 16 (8), 2011.

Adam J Berinsky, Gregory A Huber, and Gabriel S Lenz: 'Evaluating online labor markets for experimental research: Amazon.com's Mechanical Turk', *Political Analysis* 20 (3): 351–368, 2012.

Or Biran and Kathleen McKeown: 'Aggregated Word Pair Features for Implicit Discourse Relation Disambiguation', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 69–73, Sofia, Bulgaria, 2013.

Sasha Blair-Goldensohn, Kathleen McKeown, and Owen Rambow: 'Building and Refining Rhetorical-Semantic Relation Models', in: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 428–435, Rochester, New York, 2007.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio: 'Joint learning of words and meaning representations for open-text semantic parsing', in: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pp. 127–135, La Palma, Canary Islands, 2012.

Lubomir Bourdev and Jonathan Brandt: 'Robust Object Detection via Soft Cascade', in: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 236–243, Washington D.C., USA, 2005.

Andrei Z Broder: 'On the resemblance and containment of documents', in: *Proceedings of the Compression and Complexity of Sequences 1997*, pp. 21–29, Washington, DC, USA, 1997.

Andrei Z Broder, Peter Ciccolo, Marcus Fontoura, Evgeniy Gabrilovich, Vanja Josifovski, and Lance Riedel: 'Search advertising using web relevance feedback', in: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 1013–1022, Napa, California, 2008.

Carla E Brodley and Mark A Friedl: 'Identifying mislabeled training data', *Journal of Artificial Intelligence Research (JAIR)* 11: 131–167, 1999.

C Darren Brooks and Allan Jeong: 'Effects of pre-structuring discussion threads on group interaction and group performance in computer-supported collaborative argumentation', *Distance Education* 27 (3): 371–390, 2006.

Sabine Buchholz and Javier Latorre: 'Crowdsourcing Preference Tests, and How to Detect Cheating.', in: *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 3053–3056, Florence, Italy, 2011.

Michael Buhrmester, Tracy Kwang, and Samuel D Gosling: 'Amazon's Mechanical Turk a new source of inexpensive, yet high-quality, data?', *Perspectives on Psychological Science* 6 (1): 3–5, 2011.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou: 'Summarizing email conversations with clue words', in: *Proceedings of the 16th International Conference on World Wide Web*, pp. 91–100, Banff, Alberta, Canada, 2007.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou: 'Summarizing Emails with Conversational Cohesion and Subjectivity.', in: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol. 8, pp. 353–361, 2008.

Bob Carpenter, Emily Jamison, and Breck Baldwin: 'Building a Stemming Corpus: Coding Standards', `http://lingpipe-blog.com/2009/02/25/stemming-morphology-corpus-coding-standards/`, 2009.

Ben Carterette and Ian Soboroff: 'The effect of assessor error on IR system evaluation', in: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 539–546, New York, New York, 2010.

Krista Casler, Lydia Bickel, and Elizabeth Hackett: 'Separate but equal? A comparison of participants and data gathered via Amazon's MTurk, social media, and face-to-face behavioral testing', *Computers in Human Behavior* 29 (6): 2156–2160, 2013.

Richard Eckart de Castilho and Iryna Gurevych: 'A broad-coverage collection of portable NLP components for building shareable analysis pipelines', in: *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pp. 1–11, Association for Computational Linguistics and Dublin City University, Dublin, Ireland, August 2014,
Online: `http://www.aclweb.org/anthology/W14-5201`.

Kerem Çelik and T Gungor: 'A comprehensive analysis of using semantic information in text categorization', in: *Proceedings of the 2013 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–5, Albena, Bulgaria, 2013.

Joyce Yue Chai: 'Evaluation of a Generic Lexical Semantic Resource in Information Extraction.', in: *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC)*, Athens, Greece, 2000. Online proceedings.

Kam Tong Chan, Irwin King, and Man-Ching Yuen: 'Mathematical modeling of social games', in: *International Conference on Computational Science and Engineering, 2009 (CSE'09)*, Vol. 4, pp. 1205–1210, Miami, Florida, 2009.

Yee Seng Chan and Dan Roth: 'Exploiting Background Knowledge for Relation Extraction', in: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 152–160, Beijing, China, 2010.

Jesse Chandler, Gabriele Paolacci, Eyal Peer, Pam Mueller, and Kate A Ratliff: 'Using Nonnaive Participants Can Reduce Effect Sizes', *Psychological Science* 26 (7): 1131–1139, 2015.

Lydia B Chilton, John J Horton, Robert C Miller, and Shiri Azenkot: 'Task search in a human computation market', in: *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 1–9, Paris, France, 2010.

Jennifer Chu-Carroll, John Prager, Christopher Welty, Krzysztof Czuba, and David Ferrucci: 'A multi-strategy and multi-source approach to question answering', in: *Proceedings of the Eleventh Text REtrieval Conference (TREC'2002)*, pp. 124–133, Gaithersburg, Maryland, 2006.

Jacob Cohen: 'A coefficient of agreement for nominal scales', *Educational and Psychological Measurement* 20 (1): 37–46, 1960.

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun: 'Finding question-answer pairs from online forums', in: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 467–474, Singapore, 2008.

Ido Dagan, Oren Glickman, and Bernardo Magnini: 'The PASCAL Recognising Textual Entailment Challenge', in: *Machine learning challenges: Evaluating predictive uncertainty, visual object classification, and recognising textual entailment*, pp. 177–190, Springer, 2006.

Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi: 'Aggregating crowdsourced binary ratings', in: *Proceedings of the 22nd International Conference on World Wide Web*, pp. 285–294, Rio de Janeiro, Brazil, 2013.

Constantin Daniil, Mihai Dascalu, and Stefan Trausan-Matu: 'Automatic forum analysis: A thorough method of assessing the importance of posts, discussion threads and of users' involvement', in: *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, p. 37, Craiova, Romania, 2012.

Barnan Das, Narayanan C. Krishnan, and Diane J. Cook: 'Handling Imbalanced and Overlapping Classes in Smart Environments Prompting Dataset', in Katsutoshi Yada (Ed.): *Data Mining for Service*, pp. 199–219, Springer, Berlin Heidelberg, 2014.

Pradeep Dasigi, Weiwei Guo, and Mona Diab: 'Genre independent subgroup detection in online discussion threads: A pilot study of implicit attitude using latent textual semantics', in: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 65–69, 2012.

A. P. Dawid and A. M. Skene: 'Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm', *Applied Statistics* 28 (1): 20–28, 1979a.

Alexander Philip Dawid and Allan M Skene: 'Maximum likelihood estimation of observer error-rates using the EM algorithm', *Applied Statistics* pp. 20–28, 1979b.

Johannes Daxenberger, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch: 'DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data', in: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 61–66, Baltimore, Maryland, 2014, Online: `http://www.aclweb.org/anthology/P/P14/P14-5011`.

Johannes Daxenberger and Iryna Gurevych: 'Automatically Detecting Corresponding Edit-Turn-Pairs in Wikipedia', in Kristina Toutanova and Hua Wu (Eds.): *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 187–192, Association for Computational Linguistics, Stroudsburg, PA, USA, June 2014.

Ofer Dekel and Ohad Shamir: 'Vox populi: Collecting high-quality labels from a crowd', in: *Proceedings of the Twenty-Second Annual Conference on Learning Theory*, Montreal, Canada, 2009. Online proceedings.

OV Deryugina: 'Chatterbots', *Scientific and Technical Information Processing* 37 (2): 143–147, 2010.
Ellen Diep and Robert JK Jacob: 'Visualizing e-mail with a semantically zoomable interface', in: *2004 IEEE Symposium on Information Visualization (INFOVIS 2004)*, pp. 215–216, Austin, Texas, 2004.
Dominic DiPalantino and Milan Vojnovic: 'Crowdsourcing and all-pay auctions', in: *Proceedings of the 10th ACM Conference on Electronic Commerce (EC'09)*, pp. 119–128, Stanford, Calfornia, 2009.
Dmitriy Dligach and Martha Palmer: 'Reducing the need for double annotation', in: *Proceedings of the 5th Linguistic Annotation Workshop*, pp. 65–73, 2011.
Christine Doran, Guido Zarrella, and John C Henderson: 'Navigating large comment threads with CoFi', in: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 9–12, Montreal, Canada, 2012.
Julie S. Downs, Mandy B. Holbrook, Steve Sheng, and Lorrie Faith Cranor: 'Are your participants gaming the system?: Screening Mechanical Turk workers', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2399–2402, Atlanta, Georgia, 2010.
Miles Efron, Peter Organisciak, and Katrina Fenlon: 'Improving retrieval of short texts through document expansion', in: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 911–920, Portland, Oregon, 2012.
Carsten Eickhoff, Christopher G Harris, Arjen P de Vries, and Padmini Srinivasan: 'Quality through flow and immersion: gamifying crowdsourced relevance assessments', in: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 871–880, Portland, Oregon, 2012.
Carsten Eickhoff and Arjen de Vries: 'How crowdsourcable is your task', in: *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 11–14, New York, New York, 2011.
Charles Elkan: 'The Foundations of Cost-sensitive Learning', in: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 973–978, 2001.
Micha Elsner and Eugene Charniak: 'Disentangling Chat', *Computational Linguistics* 36 (3): 389–409, 2010.
Micha Elsner and Eugene Charniak: 'Disentangling chat with local coherence models', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1179–1189, 2011.
Nicolai Erbs: *Approaches to Automatic Text Structuring*, Dissertation, Technische Universität Darmstadt, Darmstadt, 2015.
Nicolai Erbs, Pedro Bispo Santos, Iryna Gurevych, and Torsten Zesch: 'DKPro Keyphrases: Flexible and Reusable Keyphrase Extraction Experiments', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.
Shai Erera and David Carmel: 'Conversation Detection in Email Systems', in: *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, pp. 498–505, Glasgow, UK, 2008.
Hui Fang: 'A Re-examination of Query Expansion Using Lexical Resources', in: *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 139–147, Columbus, Ohio, 2008.

Donghui Feng, Sveva Besana, Kirk Boydston, and Gwen Christian: 'Towards high-quality data extraction via crowdsourcing', in: *In Proceedings of the The World's First Conference on the Future of Distributed Work (CrowdConf-2010*, San Francisco, California, 2010. Electronic proceedings.

Oliver Ferschke: *The Quality of Content in Open Online Collaboration Platforms: Approaches to NLP-supported Information Quality Management in Wikipedia*, Ph.D. thesis, Technical University of Darmstadt, Darmstadt, Germany, 2014.

Oliver Ferschke, Iryna Gurevych, and Yevgen Chebotar: 'Behind the Article: Recognizing Dialog Acts in Wikipedia Talk Pages', in: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 777–786, Avignon, France, 2012.

Oliver Ferschke, Iryna Gurevych, and Marc Rittberger: 'The Impact of Topic Bias on Quality Flaw Prediction in Wikipedia', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 721–730, Sofia, Bulgaria, 2013.

Aidan Finn and Nicholas Kushmerick: 'Learning to Classify Documents According to Genre', in: *IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis*, Acapulco, Mexico, 2003. Electronic proceedings.

Jonathan G Fiscus: 'A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)', in: *Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 347–354, IEEE, Santa Barbara, California, 1997.

Max Fisher: 'Here's the e-mail trick Petraeus and Broadwell used to communicate', `https://www.washingtonpost.com/blogs/worldviews/wp/2012/11/12/heres-the-e-mail-trick-petraeus-and-broadwell-used-to-communicate/`, November 2012. Accessed: 2015-07-13.

Evgeniy Gabrilovich and Shaul Markovitch: 'Feature generation for text categorization using world knowledge', in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1048–1053, Edinburgh, Scotland, 2005.

Evgeniy Gabrilovich and Shaul Markovitch: 'Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis.', in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 7, pp. 1606–1611, Hyderabad, India, 2007.

Qin Gao and Stephan Vogel: 'Consensus versus Expertise: A Case Study of Word Alignment with Mechanical Turk', in: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 30–34, Los Angeles, California, 2010.

Emmanuel Giguet and Nadine Lucas: 'Creating discussion threads graphs with Anagora', in: *Proceedings of the 9th International Conference on Computer Supported Collaborative Learning-Volume 1*, pp. 616–620, 2009.

Eric Gilbert: 'Widespread underprovision on Reddit', in: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pp. 803–808, San Antonio, Texas, 2013.

Patricia K Gilbert and Nada Dabbagh: 'How to structure online discussions for meaningful discourse: A case study', *British Journal of Educational Technology* 36 (1): 5–18, 2005.

Howard Giles and Tania Ogay: 'Communication Accommodation Theory', in Bryan B. Whaley and Wendy Samter (Eds.): *Explaining Communication: Contemporary Theories and Exemplars*, pp. 325–344, Lawrence Erlbaum, Mahwah, New Jersey, 2007.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith: 'Part-of-speech tagging for Twitter: Annotation, features, and experiments', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 42–47, Portland, Oregon, 2011.

Vicenç Gómez, Andreas Kaltenbrunner, and Vicente López: 'Statistical analysis of the social network and discussion threads in Slashdot', in: *Proceedings of the 17th international conference on World Wide Web*, pp. 645–654, ACM, 2008.

Vicenç Gómez, Hilbert J Kappen, Nelly Litvak, and Andreas Kaltenbrunner: 'A likelihood-based framework for the analysis of discussion threads', *Proceedings of the 22nd International Conference on World Wide Web* 16 (5-6): 645–675, 2013.

Joseph K Goodman, Cynthia E Cryder, and Amar Cheema: 'Data collection in a flat world: The strengths and weaknesses of Mechanical Turk samples', *Journal of Behavioral Decision Making* 26 (3): 213–224, 2013.

Rebecca Grant: 'Reddit in 2013: 56B pageviews, 41M posts, 405M comments, and oh-so-many cats', `http://venturebeat.com/2013/12/31/reddit-in-2013-56b-pageviews-41m-posts-405m-comments-and-oh-so-many-cats/`, dec 2013. Accessed: 2015-08-28.

Jane Greenberg: 'Automatic query expansion via lexical–semantic relationships', *Journal of the American Society for Information Science and Technology* 52 (5): 402–415, 2001.

Camille Guinaudeau and Michael Strube: 'Graph-based Local Coherence Modeling', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 93–103, Sofia, Bulgaria, 2013.

Alexis Gulino: 'Reddit Saves Boy's Birthday', `http://dailycaller.com/2015/07/07/reddit-saves-boys-birthday/`, 2015. Accessed: 2015-07-07.

Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M. Meyer, and Christian Wirth: 'UBY - A Large-Scale Unified Lexical-Semantic Resource Based on LMF', in: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pp. 580–590, Avignon, France, 2012.

Dan Gusfield: *Algorithms on strings, trees and sequences: computer science and computational biology*, Cambridge University Press, 1997.

Mark Guzdial and Jennifer Turns: 'Effective discussion through a computer-mediated anchored forum', *The Journal of the Learning Sciences* 9 (4): 437–469, 2000.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten: 'The WEKA data mining software: An update', *ACM SIGKDD Explorations newsletter* 11 (1): 10–18, 2009.

Vasileios Hatzivassiloglou, Judith L Klavans, and Eleazar Eskin: 'Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning', in: *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 203–212, College Park, Maryland, 1999.

David J Hauser and Norbert Schwarz: 'It's a Trap! Instructional Manipulation Checks Prompt Systematic Thinking on "Tricky" Tasks', *SAGE Open* 5 (2), 2015. Electronic proceedings.

Paul M Healy and Krishna G Palepu: 'The fall of Enron', *Journal of Economic Perspectives* pp. 3–26, 2003.

Jim Hewitt: 'How habitual online practices affect the development of asynchronous discussion threads', *Journal of Educational Computing Research* 28 (1): 31–45, 2003.

Jim Hewitt: 'Toward an understanding of how threads die in asynchronous computer conferences', *The Journal of the Learning Sciences* 14 (4): 567–589, 2005.

Matthias Hirth, Tobias Hoßfeld, and Phuoc Tran-Gia: 'Cheat-detection mechanisms for crowdsourcing', *Technical report*, University of Würzburg, 2010.

Chu-Hong Hoi and Michael R Lyu: 'A novel log-based relevance feedback technique in content-based image retrieval', in: *Proceedings of the 12th annual ACM International Conference on Multimedia*, pp. 24–31, New York, New York, 2004.

Susan Holmes, Adam Kapelner, and Peter P Lee: 'An interactive Java statistical image segmentation system: GemIdent', *Journal of Statistical Software* 30 (10), 2009.

John J Horton: 'The condition of the Turking class: Are online employers fair and honest?', *Economics Letters* 111 (1): 10–12, 2011.

John Joseph Horton and Lydia B Chilton: 'The labor economics of paid crowdsourcing', in: *Proceedings of the 11th ACM Conference on Electronic Commerce (EC'10)*, pp. 209–218, ACM, Cambridge, Massachusetts, 2010.

Dirk Hovy, Barbara Plank, and Anders Søgaard: 'Experiments with crowdsourced re-annotation of a POS tagging data set', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 377–382, Baltimore, Maryland, 2014.

Jeff Howe: 'The rise of crowdsourcing', *Wired magazine* 14 (6): 1–4, 2006.

Jeff Howe: *Crowdsourcing: Why the power of the crowd is driving the future of business*, Random House, 2008.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani: 'Data quality from crowdsourcing: a study of annotation selection criteria', in: *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pp. 27–35, Boulder, Colorado, 2009.

Bernardo A Huberman, Daniel M Romero, and Fang Wu: 'Crowdsourcing, attention and productivity', *Journal of Information Science* 35 (6): 758–765, 2009.

Panagiotis G Ipeirotis: 'Demographics of Mechanical Turk', *Center for Digital Economy Research Working Papers* 2010. Electronic proceedings.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang: 'Quality management on Amazon Mechanical Turk', in: *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 64–67, Washington D.C., USA, 2010.

Panos Ipeirotis: 'Turker demographics vs. Internet demographics', 2009, Online: `http://behind-the-enemy-lines.blogspot.com/2009/03/turker-demographics-vs-internet.html`. Accessed 1 December 2015.

Shaili Jain and David C Parkes: 'The role of game theory in human computation systems', in: *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 58–61, Paris, France, 2009.

Emily Jamison: 'CACTUS: A User-friendly Toolkit for Semantic Categorization and Clustering in the Open Domain', in: *Proceedings of the NSF Sponsored Symposium on Semantic Knowledge Discovery, Organization and Use*, New York, New York, 2008a. Electronic proceedings.

Emily Jamison: 'Using Discourse Features for Referring Expression Generation', in: *Proceedings of the 5th Meeting of the Midwest Computational Linguistics Colloquium (MCLC)*, East Lansing, Michigan, USA, 2008b.

Emily Jamison: 'Using Online Knowledge Sources for Semantic Noun Clustering', in: *Proceedings of the Sixth Meeting of the Midwest Computational Linguistics Colloquium (MCLC)*, Bloomington, Indiana, USA, 2009. Electronic Proceedings.

Emily Jamison: 'Using Grammar Rule Clusters for Semantic Relation Classification', in: *Proceedings for the ACL Workshop ËELMS 2011: Relational Models of Semanticsẗ*, pp. 46–53, Portland, Oregon, 2011.

Emily Jamison and Iryna Gurevych: 'Headerless, Quoteless, but not Hopeless? Using Pairwise Email Classification to Disentangle Email Threads', in: *Proceedings of 9th Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pp. 327–335, Hissar, Bulgaria, 2013.

Emily Jamison and Iryna Gurevych: 'Adjacency Pair Recognition in Wikipedia Discussions using Lexical Pairs', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing*, pp. 479–488, Phuket, Thailand, 2014b.

Emily Jamison and Iryna Gurevych: 'Needle in a Haystack: Reducing the Costs of Annotating Rare-Class Instances in Imbalanced Datasets', in: *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*, pp. 244–253, Phuket,Thailand, 2014a.

Emily Jamison and Iryna Gurevych: 'Noise or additional information? Leveraging crowdsource annotation item agreement for natural language tasks', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 291–297, Lisbon, Portugal, 2015.

Emily Jamison and Dennis Mehay: 'OSU-2: Generating Referring Expressions with a Maximum Entropy Classifier', in: *Proceedings of the 5th International Natural Language Generation Conference (INLG 2008)*, pp. 196–197, Salt Fork, Ohio, USA, 2008.

Nathalie Japkowicz and Mohak Shah: *Evaluating learning algorithms: a classification perspective*, Cambridge University Press, 2011.

Matthew A Jaro: 'Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida', *Journal of the American Statistical Association* 84 (406): 414–420, 1989.

Allan C Jeong: 'The combined effects of response time and message content on growth patterns of discussion threads in computer-supported collaborative argumentation', *International Journal of E-Learning & Distance Education* 19 (1): 36–53, 2005.

Shafiq Joty, Giuseppe Carenini, Gabriel Murray, and Raymond T Ng: 'Exploiting conversation structure in unsupervised topic segmentation for emails', in: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 388–398, Cambridge, Massachusetts, 2010.

Hyun Joon Jung and Matthew Lease: 'Improving quality of crowdsourced labels via probabilistic matrix factorization', in: *Proceedings of the 4th Human Computation Workshop (HCOMP) at AAAI*, pp. 101–106, Toronto, Canada, 2012.

Phil Katz, Matt Singleton, and Richard Wicentowski: 'SWAT-MP:The SemEval-2007 Systems for Task 5 and Task 14', in: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 308–313, Prague, Czech Republic, 2007.

Bernard Kerr: 'Thread arcs: An email thread visualization', in: *IEEE Symposium on Information Visualization (INFOVIS 2003)*, pp. 211–218, Seattle, Washington, 2003.

Su Nam Kim, Li Wang, and Timothy Baldwin: 'Tagging and linking web forum posts', in: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pp. 192–202, Uppsala, Sweden, 2010.

Aniket Kittur, Ed H Chi, and Bongwon Suh: 'Crowdsourcing user studies with Mechanical Turk', in: *Proceedings of the 2008 SIGCHI Conference on Human Factors in Computing Systems*, pp. 453–456, Florence, Italy, 2008.

Aniket Kittur and Robert E Kraut: 'Harnessing the wisdom of crowds in Wikipedia: quality through coordination', in: *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, pp. 37–46, San Diego, California, 2008.

Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut: 'Crowdforge: Crowdsourcing complex work', in: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pp. 43–52, Santa Barbara, California, 2011.

Beata Beigman Klebanov and Eyal Beigman: 'Difficult Cases: From Data to Learning, and Back', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 390–396, Baltimore, Maryland, 2014.

Bryan Klimt and Yiming Yang: 'Introducing the Enron Corpus', in: *Proceedings of the Third Conference on Email and Anti-Spam (CEAS2004)*, Mountain View, California, 2004, Online: `http://bklimt.com/papers/2004_klimt_ceas.pdf`. Retrieved Dec. 9, 2015.

Aaron Michael Koblin: 'The Sheep Market', in: *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, pp. 451–452, 2009.

Moshe Koppel and Jonathan Schler: 'Exploiting Stylistic Idiosyncrasies for Authorship Attribution', in: *IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, pp. 69–72, Acapulco, Mexico, 2003.

Klaus Krippendorff: 'Estimating the reliability, systematic error and random error of interval data', *Educational and Psychological Measurement* 30 (1): 61–70, 1970.

Klaus Krippendorff: *Content analysis: An introduction to its methodology*, Sage, Beverly Hills, CA, 1980.

Carolina Kuligowska: 'Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents', *Professionals Center for Business Research* 2: 1–16, 2015.

Abhimanu Kumar and Matthew Lease: 'Modeling annotator accuracies for supervised learning', in: *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 19–22, Hong Kong, China, 2011.

John Lafferty, Andrew McCallum, and Fernando CN Pereira: 'Conditional random fields: Probabilistic models for segmenting and labeling sequence data', in: *Proceedings of the 18th International Conference on Machine Learning*, pp. 282–289, Williamstown, Massachusetts, 2001.

Vladimir I Levenshtein: 'Binary codes capable of correcting deletions, insertions, and reversals', *Soviet Physics Doklady* 10 (8): 707–710, 1966.

David D Lewis and Kimberly A Knowles: 'Threading electronic mail: A preliminary study', *Information Processing & Management* 33 (2): 209–217, 1997.

Fu-Ren Lin, Lu-Shih Hsieh, and Fu-Tai Chuang: 'Discovering genres of online discussion threads via text mining', *Computers & Education* 52 (2): 481–495, 2009a.

Jimmy Lin and Dina Demner-Fushman: 'The Role of Knowledge in Conceptual Retrieval: A Study in the Domain of Clinical Medicine', in: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 99–106, Seattle, Washington, 2006.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng: 'Recognizing Implicit Discourse Relations in the Penn Discourse Treebank', in: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 343–351, Singapore, 2009b.

Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller: 'Turkit: Tools for iterative tasks on Mechanical Turk', in: *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 29–30, Paris, France, 2009.

Caroline Lyon, Ruth Barrett, and James Malcolm: 'A theoretical basis to the automated detection of copying between texts, and its practical implementation in the Ferret plagiarism and collusion detector', *Plagiarism: Prevention, Practice and Policies* 2004,
Online: `http://uhra.herts.ac.uk/bitstream/handle/2299/2114/902216.pdf?sequence=1".` Retrieved on Dec. 9, 2015.

Daniel Marcu and Abdessamad Echihabi: 'An Unsupervised Approach to Recognizing Discourse Relations', in: *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 368–375, Philadelphia, Pennsylvania, 2002.

Hector Martinez Alonso: *Annotation of Regular Polysemy: An empirical assessment of the underspecified sense*, Ph.D. thesis, Københavns Universitet, Det Humanistiske Fakultet, 2013.

Winter Mason and Duncan J. Watts: 'Financial incentives and the "Performance of Crowds"', in: *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 77–85, Paris, France, 2009.

Margaret Mazzolini and Sarah Maddison: 'When to jump in: The role of the instructor in online discussion forums', *Computers & Education* 49 (2): 193–213, 2007.

Philip M McCarthy and Scott Jarvis: 'MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment', *Behavior Research Methods* 42 (2): 381–392, 2010.

Quinn McNemar: 'Note on the sampling error of the difference between correlated proportions or percentages', *Psychometrika* 12 (2): 153–157, 1947.

Yashar Mehdad, Alessandro Moschitti, and Fabio Massimo Zanzotto: 'Syntactic/Semantic Structures for Textual Entailment Recognition', in: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1020–1028, Los Angeles, California, 2010.

Marina Meila: 'Comparing Clusterings by the Variation of Information', in Bernhard Schölkopf and Manfred K. Warmuth (Eds.): *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, Lecture Notes in Computer Science Vol. 2777, pp. 173–187, Springer, 2003, Online: `http://dx.doi.org/10.1007/978-3-540-45167-9_14`.

Christian M. Meyer, Margot Mieskes, Christian Stab, and Iryna Gurevych: 'DKPro Agreement: An Open-Source Java Library for Measuring Inter-Rater Agreement', in: *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pp. 105–109, Dublin, Ireland, 2014.

T. Daniel Midgley, Shelly Harrison, and Cara MacNish: 'Empirical verification of adjacency pairs using dialogue segmentation', in: *Proceedings of the 7th SIGdial Workshop on Discourse and*

Dialogue, pp. 104–108, London, UK, 2009.

Rada Mihalcea and Paul Tarau: 'TextRank: Bringing Order into Texts', in: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 404–411, Barcelona, Spain, 2004.

George K. Mikros and Eleni K. Argiri: 'Investigating topic influence in authorship attribution', in: *Proceedings of the SIGIR 2007 International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN 2007)*, Amsterdam, Netherlands, 2007. Online proceedings.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller: 'Introduction to WordNet: An on-line lexical database*', *International Journal of Lexicography* 3 (4): 235–244, 1990.

Bonan Min and Ralph Grishman: 'Compensating for Annotation Errors in Training a Relation Extractor', in: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 194–203, Avignon, France, 2012.

Shachar Mirkin, Ido Dagan, and Eyal Shnarch: 'Evaluating the inferential utility of lexical-semantic resources', in: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 558–566, Athens, Greece, 2009.

Alvaro Monge and Charles Elkan: 'An efficient domain-independent algorithm for detecting approximately duplicate database records', in: *Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery*, pp. 23–29, Tucson, AZ, USA, 1997.

Araceli Moreno, Josep Lluis de la Rosa, Boleslaw K Szymanski, and José Moisés Barcenas: 'Reward System for Completing FAQs', in: *Proceeding of the 2009 conference on Artificial Intelligence Research and Development: Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence*, pp. 361–370, Amsterdam, Netherlands, 2009.

Frederick Mosteller and David Wallace: *Inference and disputed authorship: The Federalist*, Addison-Wesley, 1964.

Gabriel Murray and Giuseppe Carenini: 'Summarizing spoken and written conversations', in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 773–782, Honolulu, Hawaii, 2008.

Kevin Kyung Nam, Mark S Ackerman, and Lada A Adamic: 'Questions in, knowledge in?: A study of Naver's question answering community', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 779–788, Paris, France, 2009.

Srini Narayanan and Sanda Harabagiu: 'Question answering based on semantic structures', in: *Proceedings of the 20th international conference on Computational Linguistics (COLING)*, pp. 693–702, Geneva, Switzerland, 2004.

Vivi Nastase: 'Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation', in: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 763–772, Honolulu, Hawaii, 2008.

Stefanie Nowak and Stefan Rüger: 'How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation', in: *Proceedings of the International Conference on Multimedia Information Retrieval*, pp. 557–566, Philadelphia, Pennsylvania, 2010.

David Oleson, Alexander Sorokin, Greg P. Laughlin, Vaughn Hester, John Le, and Lukas Biewald: 'Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing.', *Human Computation* 11: 11, 2011.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith: 'Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters', in: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 380–390, Atlanta, Georgia, 2013.

Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis: 'Running experiments on Amazon Mechanical Turk', *Judgment and Decision Making* 5 (5): 411–419, 2010.

Gabriel Parent and Maxine Eskenazi: 'Clustering dictionary definitions using Amazon Mechanical Turk', in: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 21–29, Los Angeles, California, 2010.

Rebecca J Passonneau and Bob Carpenter: 'The benefits of a model of annotation', *Transactions of the Association for Computational Linguistics* 2: 311–326, 2014.

Eyal Peer, Joachim Vosgerau, and Alessandro Acquisti: 'Reputation as a sufficient condition for data quality on Amazon Mechanical Turk', *Behavior Research Methods* 46 (4): 1023–1031, 2014.

Slav Petrov, Dipanjan Das, and Ryan McDonald: 'A Universal Part-of-Speech Tagset', in: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pp. 2089–2096, Istanbul, Turkey, 2012.

Emily Pitler, Annie Louis, and Ani Nenkova: 'Automatic sense prediction for implicit discourse relations in text', in: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 683–691, Suntec, Singapore, 2009.

Barbara Plank, Dirk Hovy, and Anders Søgaard: 'Learning part-of-speech taggers with inter-annotator agreement loss', in: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 742–751, Gothenburg, Sweden, 2014.

J. Platt: 'Fast Training of Support Vector Machines using Sequential Minimal Optimization', in B. Schoelkopf, C. Burges, and A. Smola (Eds.): *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998.

Jason Pontin: 'Artificial Intelligence, With Help From the Humans', *New York Times* 2007, Online: `http://www.nytimes.com/2007/03/25/business/yourmoney/25Stream.html`. Accessed 1 December 2015.

Simone Paolo Ponzetto and Michael Strube: 'Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution', in: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 192–199, New York, New York, 2006.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber: 'The Penn Discourse TreeBank 2.0', in: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pp. 2961–2968, Marrakech, Morocco, 2008.

Alexander J Quinn and Benjamin B Bederson: 'Human computation: A survey and taxonomy of a growing field', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1403–1412, New York, New York, 2011.

Sara Radicati and Justin Levenstein: 'Email Statistics Report, 2013-2017', *Technical report*, THE RADICATI GROUP, INC., 04 2013.

Altaf Rahman and Vincent Ng: 'Coreference resolution with world knowledge', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 814–824, Portland, Oregon, 2011.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen: 'Summarizing email threads', in: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 105–108, Boston, Massachusetts, 2004.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier: 'Collecting image annotations using Amazon's Mechanical Turk', in: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 139–147, Los Angeles, California, 2010.

Vikas C. Raykar, Balaji Krishnapuram, and Shipeng Yu: 'Designing efficient cascaded classifiers: tradeoff between accuracy and cost', in: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 853–860, New York, New York, 2010a.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy: 'Learning from crowds', *The Journal of Machine Learning Research* 11: 1297–1322, 2010b.

Umaa Rebbapragada, Lukas Mandrake, Kiri L Wagstaff, Damhnait Gleeson, Rebecca Castano, Steve Chien, and Carla E Brodley: 'Improving onboard analysis of Hyperion images by filtering mislabeled training data examples', in: *Proceedings of the 2009 IEEE Aerospace Conference*, pp. 1–9, Big Sky, Montana, USA, 2009.

Dennis Reidsma and Jean Carletta: 'Reliability measurement without limits', *Computational Linguistics* 34 (3): 319–326, 2008.

Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson: 'Who Are the Crowdworkers?: Shifting Demographics in Mechanical Turk', in: *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, pp. 2863–2872, 2010.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson: 'A simplest systematics for the organization of turn-taking for conversation', *Language* pp. 696–735, 1974.

Emanuel A. Schegloff: 'On the organization of sequences as a source of "coherence" in talk-in-interaction', *Conversational organization and its development* 38: 51–77, 1990.

Emanuel A. Schegloff and Harvey Sacks: 'Opening up closings', *Semiotica* 8 (4): 289–327, 1973.

Robert P Schumaker, Ying Liu, Mark Ginsburg, and Hsinchun Chen: 'Evaluating mass knowledge acquisition using the ALICE chatterbot: The AZ-ALICE dialog system', *International Journal of Human-Computer Studies* 64 (11): 1132–1140, 2006.

Roy Schwartz, Omri Abend, Roi Reichart, and Ari Rappoport: 'Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 663–672, Association for Computational Linguistics, Portland, Oregon, 2011.

John Scott: *Social network analysis*, Sage, 2012.

D Sculley and Gordon V Cormack: 'Filtering Email Spam in the Presence of Noisy User Feedback.', in: *Conference on Email and Anti-spam (CEAS)*, Mountain View, California, 2008. Online proceedings.

Jangwon Seo, W. Bruce Croft, and David A. Smith: 'Online community search using thread structure', in: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 1907–1910, Hong Kong, China, 2009.

Jangwon Seo, W Bruce Croft, and David A Smith: 'Online community search using conversational structures', *Information Retrieval* 14 (6): 547–571, 2011.

Dan Shen and Mirella Lapata: 'Using Semantic Roles to Improve Question Answering', in: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 12–21, Prague, Czech Republic, 2007.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis: 'Get another label? Improving data quality and data mining using multiple, noisy labelers', in: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 614–622, Las Vegas, Nevada, 2008.

S.K. Shevade, S.S. Keerthi, C. Bhattacharyya, and K.R.K. Murthy: 'Improvements to the SMO Algorithm for SVM Regression', *IEEE Transactions on Neural Networks* 11 (5): 1188–1193, 1999.

Lokesh Shrestha and Kathleen McKeown: 'Detection of question-answer pairs in email conversations', in: *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, p. 889, Geneva, Switzerland, 2004.

M Silberman, Lilly Irani, and Joel Ross: 'Ethics and tactics of professional crowdwork', *XRDS: Crossroads, The ACM Magazine for Students* 17 (2): 39–43, 2010.

Marc A Smith and Andrew T Fiore: 'Visualization components for persistent conversations', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 136–143, New York, New York, 2001.

Padhraic Smyth, Usama Fayyad, Michael Burl, Pietro Perona, and Pierre Baldi: 'Inferring ground truth from subjective labelling of Venus images', *Advances in Neural Information Processing Systems* pp. 1085–1092, 1995.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng: 'Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks', in: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 254–263, Honolulu, Hawaii, 2008.

Benjamin Snyder and Martha Palmer: 'The English all-words task', in Rada Mihalcea and Phil Edmonds (Eds.): *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 41–43, Association for Computational Linguistics, Barcelona, Spain, July 2004.

Alexander Sorokin and David Forsyth: 'Utility data annotation with Amazon Mechanical Turk', *Urbana* 51 (61): 820, 2008.

Jon Sprouse: 'A validation of Amazon Mechanical Turk for the collection of acceptability judgments in linguistic theory', *Behavior Research Methods* 43 (1): 155–167, 2011.

Efstathios Stamatatos: 'Plagiarism Detection Using Stopword N-grams', *Journal of the American Society for Information Science and Technology* 62 (12): 2512–2527, 2011.

Neil Stewart, Christoph Ungemach, Adam JL Harris, Daniel M Bartels, Ben R Newell, Gabriele Paolacci, and Jesse Chandler: 'The Average Laboratory Samples a Population of 7,300 Amazon Mechanical Turk Workers', *Judgment and Decision Making* 10 (5): 479–491, 2015.

Carlo Strapparava and Rada Mihalcea: 'SemEval-2007 Task 14: Affective Text', in: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 70–74, Prague, Czech Republic, 2007.

strategyeye.com: 'WhatsApp surpasses global SMS volume', `http://digitalmedia.strategyeye.com/article/BSVgW3sszU/2014/01/21/whatsapp_surpasses_global_sms_volume/`, 2014. Accessed: 2015-06-17.

Alexander Strehl: *Relationship-based clustering and cluster ensembles for high-dimensional data mining*, Ph.D. thesis, The University of Texas at Austin, 2002. PhD Thesis.

Jan Svennevig: *Getting acquainted in conversation. A study of initial interactions*, Pragmatics & Beyond New Series, 1999.

Wei Tang and Matthew Lease: 'Semi-supervised consensus labeling for crowdsourcing', in: *SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR)*, pp. 36–41, Beijing, China, 2011.

Alexey Tarasov, Sarah Jane Delany, and Brian Mac Namee: 'Dynamic estimation of worker reliability in crowdsourcing for regression tasks: Making it work', *Expert Systems with Applications* 41 (14): 6190–6210, 2014.

Mildred C Templin: *Certain language skills in children; their development and interrelationships*, University of Minnesota Press, 1957.

Christian Thiel: 'Classification on soft labels is robust against label noise', in: *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 65–73, Wellington, New Zealand, 2008.

Matthew JW Thomas: 'Learning within incoherent structures: The space of online discussion forums', *Journal of Computer Assisted Learning* 18 (3): 351–366, 2002.

Julie Tibshirani and Christopher D. Manning: 'Robust Logistic Regression using Shift Parameters', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 124–129, Association for Computational Linguistics, Baltimore, Maryland, 2014, Online: `http://www.aclweb.org/anthology/P14-2021`.

Mark Twain: 'A Telephonic Conversation', *The Atlantic* 1880, Online: `http://www.theatlantic.com/magazine/archive/1880/06/a-telephonic-conversation/306078/`. Accessed 1 December 2015.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova: 'Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion', *Information Processing & Management* 43 (6): 1606–1618, 2007.

Gina Danielle Venolia and Carman Neustaedter: 'Understanding sequence and reply relationships within email conversations: A mixed-model visualization', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 361–368, ACM, 2003.

Fernanda B Viégas, Scott Golder, and Judith Donath: 'Visualizing email content: Portraying relationships from conversational histories', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 979–988, New York, New York, 2006.

Paul A. Viola and Michael J. Jones: 'Rapid Object Detection using a Boosted Cascade of Simple Features', in: *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,

pp. 511–518, Kauai, Hawaii, 2001.

Luis Von Ahn: 'Games with a purpose', *Computer* 39 (6): 92–94, 2006.

Ellen M Voorhees: 'Query expansion using lexical-semantic relations', in: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR'94)*, pp. 61–69, Dublin, Ireland, 1994.

Stephen Wan and Kathy McKeown: 'Generating overview summaries of ongoing email thread discussions', in: *Proceedings of the 20th international conference on Computational Linguistics (COLING)*, p. 549, Geneva, Switzerland, 2004.

Hongning Wang, Chi Wang, ChengXiang Zhai, and Jiawei Han: 'Learning online discussion structures by conditional random fields', in: *Proceedings of the 34th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 435–444, Beijing, China, 2011a.

Jing Wang, Siamak Faridani, and P Ipeirotis: 'Estimating the completion time of crowdsourced tasks using survival analysis models', in: *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM 2011)*, pp. 31–34, New York, New York, 2011b.

Li Wang, Su Nam Kim, and Timothy Baldwin: 'The Utility of Discourse Structure in Forum Thread Retrieval', in: *Information Retrieval Technology*, pp. 284–295, Springer, 2013.

Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin: 'Predicting thread discourse structure over technical web forums', in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 13–25, Edinburgh, UK, 2011c.

Li Wang, Diana McCarthy, and Timothy Baldwin: 'Predicting Thread Linking Structure by Lexical Chaining', in: *Proceedings of the Australasian Language Technology Association Workshop 2011*, p. 76, Canberra, Australia, 2011d.

Yi-Chia Wang, Mahesh Joshi, William W Cohen, and Carolyn Penstein Rosé: 'Recovering Implicit Thread Structure in Newsgroup Style Conversations.', in: *Proceedings of the International Conference on Weblogs and Social Media*, pp. 152–160, Seattle, Washington, 2008.

Yi-Chia Wang and Carolyn P. Rosé: 'Making conversational structure explicit: identification of initiation-response pairs within online discussions', in: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 673–676, Los Angeles, California, 2010.

Martin Warren: *Features of naturalness in conversation*, Vol. 152, John Benjamins Publishing, 2006.

Fabian L Wauthier and Michael I Jordan: 'Bayesian bias mitigation for crowdsourcing', in: *Advances in Neural Information Processing Systems*, pp. 1800–1808, 2011.

Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie: 'The multidimensional wisdom of crowds', in J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta (Eds.): *Advances in Neural Information Processing Systems (NIPS)*, pp. 2424–2432, Vancouver, Canada, 2010.

Peter Welinder and Pietro Perona: 'Online crowdsourcing: rating annotators and obtaining cost-effective labels', in: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 25–32, San Francisco, California, 2010.

Howard T Welser, Eric Gleave, Danyel Fisher, and Marc Smith: 'Visualizing the signatures of social roles in online discussion groups', *Journal of Social Structure* 8 (2): 1–32, 2007.

Tim Weninger, Xihao Avi Zhu, and Jiawei Han: 'An exploration of discussion threads in social news sites: A case study of the Reddit community', in: *2013 IEEE/ACM International Conference on*

Advances in Social Networks Analysis and Mining (ASONAM), pp. 579–583, Niagara Falls, ON, Canada, 2013.

Jacob Whitehill, Ting fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo: 'Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise', in Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta (Eds.): *Advances in Neural Information Processing Systems 22*, pp. 2035–2043, Curran Associates, Inc., 2009.

Wikipedia: 'CrowdFlower — Wikipedia, The Free Encyclopedia', 2015e, Online: https://en.wikipedia.org/w/index.php?title=CrowdFlower&oldid=669927484. [Online; accessed 7-August-2015].

Wikipedia: 'Crowdsourcing — Wikipedia, The Free Encyclopedia', 2015c, Online: https://en.wikipedia.org/w/index.php?title=Crowdsourcing&oldid=674652467. [Online; accessed 7-August-2015].

Wikipedia: 'John Harrison — Wikipedia, The Free Encyclopedia', 2015b, Online: https://en.wikipedia.org/w/index.php?title=John_Harrison&oldid=668210657. [Online; accessed 7-August-2015].

Wikipedia: 'Longitude rewards — Wikipedia, The Free Encyclopedia', 2015a, Online: https://en.wikipedia.org/w/index.php?title=Longitude_rewards&oldid=672374940. [Online; accessed 7-August-2015].

Wikipedia: 'The Turk — Wikipedia, The Free Encyclopedia', 2015d, Online: https://en.wikipedia.org/w/index.php?title=The_Turk&oldid=656467570. [Online; accessed 7-August-2015].

William E Winkler: 'String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.', in: *Proceedings of the Section on Survey Research Methods*, pp. 354–359, Alexandria, Virginia, 1990.

Michael J Wise: 'YAP3: Improved detection of similarities in computer program and other texts', *Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education* 28 (1): 130–134, 1996.

Weining Wu, Yang Liu, Maozu Guo, Chunyu Wang, and Xiaoyan Liu: 'A probabilistic model of active learning with multiple noisy oracles', *Neurocomputing* 118: 253–262, 2013.

Yejun Wu and Douglas W Oard: 'Indexing emails and email threads for retrieval', in: *Proceedings of the 28th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 665–666, New York, New York, 2005.

Jiang Yang, Lada A Adamic, and Mark S Ackerman: 'Crowdsourcing and knowledge sharing: Strategic user behavior on taskcn', in: *Proceedings of the 9th ACM Conference on Electronic Commerce (EC'08)*, pp. 246–255, Chicago, Illinois, 2008.

Tae Yano, Philip Resnik, and Noah A Smith: 'Shedding (a thousand points of) light on biased language', in: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 152–158, Los Angeles, California, 2010.

Jen-Yuan Yeh and Aaron Harnly: 'Email thread reassembly using similarity matching', in: *Proceedings of the Third Conference on Email and Anti-Spam (CEAS2006)*, Mountain View, California, 2006, Online: http://academiccommons.columbia.edu/download/fedora_content/download/ac:162861/CONTENT/yeh_harnly_06.pdf. Retrieved on Dec. 9, 2015.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak: 'Question Answering Using Enhanced Lexical Semantic Models', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 1744–1753, Sofia, Bulgaria, 2013.

Man-Ching Yuen, Ling-Jyh Chen, and Irwin King: 'A survey of human computation systems', in: *International Conference on Computational Science and Engineering, 2009 (CSE'09)*, Vol. 4, pp. 723–728, Miami, Florida, 2009.

Man-Ching Yuen, Irwin King, and Kwong-Sak Leung: 'A survey of crowdsourcing systems', in: *Proceedings of the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International conference on Social Computing (SocialCom)*, pp. 766–773, Boston, Massachusetts, 2011a.

Man-Ching Yuen, Irwin King, and Kwong-Sak Leung: 'Task matching in crowdsourcing', in: *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pp. 409–412, Dalian, China, 2011b.

G Udny Yule: 'On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship', *Biometrika* 30 (3/4): 363–390, 1939.

Omar F. Zaidan and Chris Callison-Burch: 'Crowdsourcing Translation: Professional Quality from Non-Professionals', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1220–1229, Portland, Oregon, 2011.

Weizhong Zhu, Robert B Allen, and Min Song: 'TREC 2005 Enterprise Track Results from Drexel', in: *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, Gaithersburg, Maryland, 2005.

Jonathan Zittrain: 'The Internet Creates a New Kind of Sweatshop', *Newsweek* 2009, Online: `http://www.newsweek.com/internet-creates-new-kind-sweatshop-75751`. Accessed 1 December 2015.

Nathan Zukoff: 'Demographics of the Largest On-demand Workforce', jan 2014, Online: `http://www.crowdflower.com/blog/2014/01/demographics-of-the-largest-on-demand-workforce`. [Online; accessed 7-August-2015].

Appendix

A Corpora with crowdsource annotation item agreement

In this section, we provide samples of the datasets used in Chapter 4, along with their crowdsource annotation α item agreement.

Agr	Label	Blog Text
1.0	2.0	She's not convincing anyone , in other words , she's simply asking them to ... "fall in line."
1.0	2.0	Now that she 's running for President , Clinton has changed her tune .
1.0	1.0	Hill wins Ohio and RI. "We're going all the way ! " Crowd shouts , "Yes , she will."
0.3	1.8	McCain apologists will argue that Sarah Palin was not a member of this group .
1.0	1.0	Among women , Clinton leads 64 % to 31 % .
0.3	1.2	Petraeus : I am not using the word "brief" or the word "pause."
0.3	1.8	I fought in the Senate for the most extensive ethics reform since Watergate .
1.0	1.0	It 's not the news media 's job to make a judgment about whether they were right to do so .
0.3	1.2	These guys can run for president but they ca n't be Secretary of the Treasury.Matthews :
1.0	1.0	Might that change in the coming days if the bailout package passes ?
0.3	1.2	Within minutes of Thompson's exit , the Romney campaign had a reaction statement up on its web site .
0.3	2.8	They died because of the Bush administration's hubris .
1.0	1.0	SIMMONS : I will quibble with one point that was just made .
1.0	1.0	The original Taliban had mostly been displaced as refugees into Pakistan .
1.0	1.0	" I do n't blame the Army for our son 's death , " Nancy says .
0.3	1.4	So it looks like there are votes that were properly counted on Election Night , but are missing right now .
0.3	1.2	History might show that General David Patreaus as a great American military hero .
0.3	1.4	In the city of Tyre , too , posters showing young men killed in training exercises are cropping up .
1.0	1.0	First , like McConnell , he just won re-election and won't have the distraction of personal campaigning.
0.3	2.2	His "I led for patriotism , not for profit" line is a slap in the face to business .
0.3	1.8	If the answers didn't reflect his views , why didn't he change them when he "jotted some notes" on it ?
0.3	1.8	Exit question : Which racial stereotype is the Raines ad supposedly playing on ?
0.3	1.2	47 percent now favor "immediate withdrawal of U.S. forces , " a 12-point rise since March .

Table 1: Easy Case biased language text from Yano2010.

Agr	Label	Blog Text
-0.4	2.2	And that can very easily be resolved by Senator Obama , by Mrs. Obama , by Mr. Ayers and by Ms. Dohrn . ”
-0.4	2.0	On Monday , he said the economy was fundamentally sound , and he was fundamentally wrong .
-0.4	2.2	*** Flashback : An unhinged moonbat threw a shoe at Richard Perle in 2005 during a speech in Portland .
-0.4	1.8	ThinkProgress has gladly taken up the McCain challenge .
-0.4	2.0	Without Americans getting killed , CBS does n’t see a story.And
-0.2	2.0	If Murtha loses , this will end up being remembered as the Democratic version of a “ Macaca Moment . ”
-0.4	1.8	Whether Palin truly understands the role of the vice president has been repeatedly called into question .
-0.4	1.8	But now McCain seems to be fairly certain Obama is a socialist .
-0.4	1.8	But I ’ve got to say , she ’s opposed - like John McCain is - to equal pay for equal work .
-0.4	1.8	And McCain’s would cut taxes , cut the overall tax burden .
-0.4	2.2	We ought to go through because they’re not telling the truth , there’s no risk , it ought to be done .
-0.4	2.0	He has said that he really liked the ideas of the Republicans over the last 10-15 years .
-0.4	1.8	Why would Palin endorse anyone , though ?
-0.4	2.0	Earlier this week , Rep. John Shadegg -LRB- R-AZ -RRB- called it the “2008 version of the Boston Tea Party.”
-0.4	1.8	But the Iranian leadership does far more than issue vile insults .
-0.4	1.8	Beautiful words can not make our lives better … Do n’t hope for a better life – vote for one. –
-0.4	1.8	But one of the most troubling parts of the memo concerns the office’s close relationship with lobbyists .
-0.4	2.2	MCCAIN : We would make them shamed into it .
-0.4	2.2	And it includes saying and doing just about anything to win .
-0.4	2.0	After all , Obama is the first black candidate who has a real shot at winning .
-0.4	2.2	Bush smiled and made his usual quips , and many of the reporters played the game and did not press him hard .
-0.2	1.9	I’m sure Fairey did this , and I’ll tell you why .

Table 2: Hard Case biased language text from YANO2010.

Stem Pairs: Negative						Stem Pairs: Positive					
Original	**Stemmed**	**Agr**	**Original**	**Stemmed**	**Agr**	**Original**	**Stemmed**	**Agr**	**Original**	**Stemmed**	**Agr**
honorably	honor	-0.1	whimsy	whim	-0.6	railbikes	railbike	-0.6	riders	rider	1.0
droves	droves	-0.6	samplings	sample	-0.1	backdated	backdate	-0.1	eh	eh	-0.1
unanimity	unanimity	-0.6	alleyways	alley way	-0.1	vulgarity	vulgar	1.0	narrowing	narrow	1.0
unsparing	sparing	-0.1	sons	DELETED	-0.1	groomer	groom	1.0	convoy	convoy	1.0
drubbed	DELETED	-0.6	jokester	joke	-0.1	magnitude	magnitude	1.0	tolerance	tolerate	-0.1
jokester	jokes	-0.1	reponse	reponse	-0.1	agitating	agitate	1.0	patches	patch	-0.1
reponse	repond	-0.1	hauteur	haute	-0.6	lifer	life	1.0	passively	passive	1.0
declining	declinie	-0.1	nascent	DELETED	-0.1	samba	samba	1.0	assuming	assume	1.0
injuction	injuction	-0.6	injuction	DELETED	-0.1	repayed	repay	1.0	femme	femme	1.0
injuction	injuct	-0.1	zingy	zingy	-0.1	confusion	confuse	1.0	moored	moor	1.0
foie	DELETED	-0.1	bogyman	DELETED	-0.6	lawmaker	law maker	1.0	currently	current	1.0
bogyman	bogyman	-0.6	voluntary	voluntary	-0.6	oppress	oppress	1.0	preachers	preacher	1.0
oozing	ooz	-0.1	railbikes	rail bike	-0.1	intact	intact	1.0	leotards	leotard	1.0
bunker	bunker	-0.6	eh	DELETED	-0.1	sods	sod	1.0	glorying	glory	1.0
flyovers	fly over	-0.1	pianist	pianist	-0.1	reponse	DELETED	-0.6	gastritis	gastritis	-0.1
shek	shek	-0.1	cree	DELETED	-0.6	hauteur	hauteur	-0.6	disposes	dispose	1.0
expelled	expell	-0.6	tugboats	tugnboat	-0.1	rabid	rabid	1.0	exorcism	exorcise	-0.6
weeks	weeks	-0.1	walkie	walkie	-0.6	rumpus	rumpus	1.0	supposing	suppose	1.0
chastity	chastity	-0.6	callups	DELETED	-0.1	ski	ski	1.0	smaller	small	-0.1
boxloads	box load	-0.1	swingeing	swing	-0.1	unimpeded	impeded	-0.1	vanish	vanish	1.0
litle	litle	-0.1	anchorage	anchorage	-0.1	wrappings	wrapping	1.0	coercive	coerce	1.0
faceted	face	-0.1	overhung	hung	-0.6	nascent	nascent	-0.1	loyalists	loyalist	1.0
proudest	prouder	-0.6	former	form	-0.6	manor	manor	1.0	jostle	jostle	1.0
plaintive	plain	-0.1	former	fore	-0.1	weevil	weevil	1.0	users	user	1.0
former	former	-0.6	residence	reside	-0.6	bouts	bout	1.0	workbook	work book	1.0
creation	creation	-0.1	residence	residence	-0.6	subsist	subsist	1.0	compere	DELETED	-0.1
residence	resident	-0.1	ost	ost	-0.6	class	class	1.0	riles	rile	1.0
orderlies	orderlie	-0.1	unrolled	rolled	-0.1	didn	DELETED	-0.1	zingy	zing	-0.1
sedative	sedative	-0.1	heritable	herit	-0.1	peekaboo	peekaboo	-0.6	refrain	refrain	1.0
inimical	DELETED	-0.1	palimony	DELETED	-0.1	batters	batter	1.0	describes	describe	1.0
palimony	alimony	-0.1	winning	winn	-0.1	shabbily	shabby	-0.1	heeded	heed	1.0
vetch	DELETED	-0.1	hacerse	DELETED	-0.1	redesigns	redesign	1.0	prosper	prosper	1.0
hacerse	hacer	-0.1	yester	DELETED	-0.1	granaries	granary	1.0	frogs	frog	1.0
preserves	preserves	-0.1	songbooks	song book	-0.1	soaring	soar	-0.1	cree	cree	-0.6
liftoffs	lift off	-0.1	whopping	whop	-0.1	affective	affect	1.0	ku	DELETED	-0.6

Table 3: Stemming word pairs with agreement and class, from Carp2009.

Text	Hypothesis	Agr
A former petty thief who converted and founded his own church, Silva is a devoted jailhouse preacher who claims to have ended 11 prison rebellions in recent years.	Silva was once a murderer.	0.3
Most Americans are familiar with the Food Guide Pyramid– but a lot of people don't understand how to use it.	Most Americans have not heard of the Food Guide Pyramid.	0.0
About 85 percent of Danes belong to the state Evangelical Lutheran Church, though just 5 percent attend church services regularly.	85 percent of Danes attend church services regularly.	-0.1
A Health Ministry official said 68 people were killed and 30 wounded in the blast shortly after 10 a.m. in Baquba, an often violent town 65 km north of Baghdad.	Baghdad is north of Baquba.	-0.1
The 69-page report is also the first major product of the Betsy Lehman Center for Patient Safety and Medical Error Reduction.	The 69-page report is the first major product of medical errors.	-0.1
Japan's voter turnout was just over 56 percent for the Upper House elections.	Less than half of the eligible Japanese voters participated in the vote.	0.6
North Korean refugees had been gathering in southern Ho Chi Minh City, formerly Saigon, after trickling over the border from China for months.	Ho Chi Minh City is now called Saigon.	-0.1
Iraqi militants have repeatedly used terrorist attacks to try to force governments to withdraw from the U.S.-led occupation force.	Iraqi militants were forced to withdraw from the U.S.-led occupation force.	-0.1
Four Venezuelan firefighters who were traveling to a training course in Texas were killed when their sport utility vehicle drifted onto the shoulder of a highway and struck a parked truck.	Four firefighters were killed while saving a man stuck in a burning building.	0.6
All but 11 of the 107 patients and all of the employees had been notified by late yesterday to come to the hospital for an evaluation and antibiotics.	11 patients out of the 107 were invited to the hospital for a check-up and antibiotics.	0.0
Mrs. Lane, who has been a Director since 1989, is Special Assistant to the Board of Trustees and to the President of Stanford University.	Mrs. Lane is the president of Stanford University.	0.0
Lyon acclaims itself to be the gastronomic capital of France.	Lyon is the capital of France.	0.3
Israel captured the Gaza Strip and West Bank in the 1967 Mideast war.	The Gaza Strip and West Bank were captured by Israel in the 1976 war.	0.0
Fighters loyal to Moqtada al-Sadr shot down a U.S. helicopter Thursday in the holy city of Najaf.	A U.S. helicopter flew loyalists of Moqtada al-Sadr.	0.3

Table 4: Negative RTE examples, with agreement. The text is from Pascal Rte-1, and the labels are from RteAnno.

Text	Hypothesis	Agr
The American Consul-General in Jerusalem, John Hearst, and an high-level Israeli officer in military uniform were among the dignitaries.	There were American and Israeli dignitaries present.	0.3
Newspapers said that every independent candidate spent between 200 thousand and 500 thousand dollars on his election campaign.	Independent candidates must spend $50,000 on their election campaigns.	-0.1
The plane will be prepared for President Arafat's flight to Paris on Wednesday, after he confirmed that he would allocate his first international trip, to France.	Arafat will devote his first international trip to France.	0.3
The opinion poll was conducted on the sixth and seventh of October, and included a cross section of 861 adults with a margin of error estimated at 4%.	The poll was carried out on the 6th and 7th of October.	1.0
5-year-old, family prepare for risky marrow transplant.	5-year-old, family prepare for risky bone marrow transplant.	0.6
if you are at risk of heart problems, it is now recommended that you talk to your doctor about taking aspirin to prevent a first heart attack.	Aspirin use lowers risk of heart disease.	-0.1
Five other soldiers have been ordered to face courts-martial.	Five other soldiers have been demanded to face courts-martial.	0.6
Black holes can lose mass by radiating energy in the form of "Hawking radiation".	Black holes can regain some of their mass by radiating energy.	-0.1
The girl's mother and grandmother have been charged with conspiracy to commit murder.	The girl's mother and grandmother face charges of conspiracy to commit murder.	0.6
Weekly numbers also showed that refinancing activity sailed to its highest level since January and jobless claims dropped to the lowest level since the recession began in January of 2001.	Jobless claims fall to lowest level since January 2001.	0.3
Bush returned to the White House late Saturday while his running mate was off campaigning in the West.	Bush left the White House.	-0.1
The prosecutor told the court that the incident had caused "distress" to one of the children.	The prosecutor told the court that "distress" in one of the children is ascribed to the incident.	1.0
Mr Arafat's opponents still blame him for the mounting lawlessness in the Palestinian territories.	Mr Arafat's opponents accuse him of being responsible for the mounting lawlessness in the Palestinian territories.	0.6
On 2 February 1990, at the opening of Parliament, he declared that apartheid had failed and that the bans on political parties, including the ANC, were to be lifted.	Apartheid in South Africa was abolished in 1990.	-0.1

Table 5: Positive RTE examples, with agreement. The text is from Pascal Rte-1, and the labels are from RteAnno.

Token	Labels	Agr	Token	Labels	Agr
hehehe	PRT,PRT,X,PRT,PRT	0.5	@USER	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
its	PRON,DET,VERB,PRON,PRON	0.2	-lol	NUM,X,X,NUM,NOUN	0.1
gonna	PRT,VERB,PRT,VERB,VERB	0.3	im	VERB,PRON,PRON,PRON,PRON	0.5
b	VERB,PRT,PRT,VERB,VERB	0.3	jk	PRT,PRT,PRT,PRT,PRT	1.0
a	DET,DET,DET,DET,DET	1.0	ahah	PRT,PRT,PRT,PRT,PRT	1.0
good	ADJ,ADJ,ADJ,ADJ,ADJ	1.0	itss	PRON,PRON,PRT,PRON,PRT	0.3
day	NOUN,NOUN,NOUN,NOUN,NOUN	1.0	okayy	PRT,NOUN,NOUN,ADJ,ADJ	0.1
			…	.,.,.,PRT,PRT	0.3
@USER	NOUN,NOUN,NOUN,NOUN,NOUN	1.0			
Lmao	PRT,PRT,PRT,PRT,PRT	1.0	#LebronShould	X,X,X,X,X	1.0
oh	PRT,PRT,.,PRT,PRT	0.5	know	NOUN,VERB,VERB,VERB,VERB	0.5
ok	PRT,NOUN,ADJ,CONJ,NOUN	0.0	his	PRON,ADJ,PRON,PRON,PRON	0.5
i	PRON,PRON,PRON,PRON,PRON	1.0	only	ADJ,ADJ,ADV,ADJ,ADP	0.2
was	VERB,VERB,VERB,VERB,VERB	1.0	championship	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
like	CONJ,VERB,PRT,ADP,ADP	0.0	is	VERB,VERB,VERB,VERB,VERB	1.0
where	ADP,ADV,ADP,PRON,ADV	0.1	’	.,.,.,.,.	1.0
is	VERB,VERB,VERB,VERB,VERB	1.0	slam	VERB,NOUN,ADJ,ADJ,ADJ	0.2
she	PRON,PRON,PRON,NOUN,NOUN	0.3	dunk	VERB,NOUN,VERB,VERB,NOUN	0.3
at	ADP,ADP,ADP,ADV,ADP	0.5	’.	.,.,.,.,.	1.0
			@USER	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
I	PRON,PRON,PRON,PRON,PRON	1.0			
wanna	VERB,VERB,VERB,ADJ,VERB	0.5	Journalists	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
go	VERB,VERB,VERB,VERB,VERB	1.0	and	CONJ,CONJ,CONJ,CONJ,CONJ	1.0
to	VERB,CONJ,VERB,ADP,ADP	0.1	Social	NOUN,NOUN,ADJ,ADJ,ADJ	0.3
a	DET,DET,DET,DET,DET	1.0	Media	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
bar	NOUN,NOUN,NOUN,NOUN,NOUN	1.0	experts	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
….	.,.,.,.,.	1.0	alike	ADJ,ADJ,ADP,ADV,ADJ	0.2
not	ADJ,ADV,ADP,ADP,ADV	0.1	will	VERB,VERB,VERB,ADV,VERB	0.5
to	NOUN,ADP,ADP,ADP,ADP	0.5	appreciate	VERB,VERB,VERB,ADV,NOUN	0.2
drink	VERB,VERB,VERB,ADV,VERB	0.5	this	ADP,PRON,ADJ,NOUN,ADP	0.0
tho	ADV,ADV,CONJ,DET,VERB	0.0	spoof	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
…..	.,.,.,.,.	1.0	out	ADP,ADV,ADP,VERB,ADV	0.1
just	ADJ,ADV,ADJ,ADV,ADV	0.3	of	ADP,ADP,ADP,ADP,ADP	1.0
to	VERB,ADP,ADP,ADP,ADP	0.5	Dallas	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
get	VERB,VERB,VERB,VERB,VERB	1.0	:	.,.,.,.,.	1.0
out	ADP,ADV,ADV,ADP,VERB	0.1	URL	NOUN,NOUN,NOUN,NOUN,NOUN	1.0
the	DET,DET,DET,DET,DET	1.0			
house	NOUN,NOUN,NOUN,NOUN,NOUN	1.0			

Table 6: Tweets, with crowdsourced POS labels and item agreement. The text is from Gimbel2011 and the labels are from GimbelAnno.

Hard Cases			Easy Cases		
Agr	**Label**	**Headline**	**Agr**	**Label**	**Headline**
-0.6	26.0	Ice storm smacks roads, power	0.4	17.0	5000 years on but couple still hugging
-0.6	34.5	Two Hussein allies are hanged, Iraqi official says	0.4	12.5	At New OZZFEST, Freedom Ain't Free
-0.3	16.4	Democrats plot Bush troop increase censure	0.7	5.0	Will Rob Cohen Direct Third 'Mummy'?
-0.1	11.7	Really?: The claim: the pill can make you put on weight	0.4	7.0	Shareholders sue Apple
-0.6	48.4	Storms kill, knock out power, cancel flights	0.7	0.0	Sights and sounds from CES
-0.3	28.1	Vaccine mandate upsets legislators	0.4	5.0	Defense to challenge Russert's credibility
-0.6	33.5	Iran says it will strike US interests if Attacked	0.4	7.4	5 money makeovers
-0.5	65.5	Aquarium puts ailing beluga whale to sleep	0.7	5.0	Federer handed tough Aussie draw
-0.6	45.5	Israelis retaliate after attack by Lebanese Army	0.4	3.0	News analysis: Iranian boast is put to test
-0.7	35.6	Panel issues bleak report on climate change	0.7	0.1	EU will urge China to go green
-0.5	40.5	Deadly bird flu confirmed in British Turkeys	0.7	4.5	Palestinian factions to resume talks
-0.4	28.5	Closings and cancellations top advice on flu outbreak	0.4	1.0	Schuey sees Ferrari unveil new car
-0.4	23.0	CIA leak trial summary	0.7	10.0	Ozzy, a Hero for the hard-rocking masses
-0.2	37.0	A police state? The issues	0.4	2.7	BB star Jackson denies Goody comments
-0.5	39.0	Too little sleep may mean too fat kids	0.7	2.0	Merck: Gardasil may fight more strains
-0.6	32.5	Outcry at N Korea 'nuclear test'	1.0	0.0	Microsoft, Sony, we have a problem
-0.7	61.7	Trucks swallowed in subway collapse	0.7	10.0	17th-Century Remedy; 21st-Century Potency
-0.4	52.9	Two detained in body parts mailing	0.4	2.7	Ganguly handed India squad call-up
-0.5	73.4	Iraq car bombings kill 22 People, wound more than 60	0.7	1.2	Inter Milan set Serie A win record
-0.2	93.5	Bathing mom awakes to find baby dead	0.4	2.4	Bears fan loses bet and changes name
-0.7	41.0	Russia plans major military build-up	0.7	10.0	Turner pays for Boston "bombing"
-0.5	35.1	Asian nations urge Myanmar reform	0.7	0.0	Virtual 'American Idol' hits right notes
-0.5	72.4	Teacher charged with sex assault	0.7	2.0	Dance movie takes over No. 1
-0.7	50.5	Archaeologists find remains of couple locked in a hug	0.7	6.5	US Airways boosts bid for Delta
-0.4	26.9	Building a memorial to a son, one child at a time	0.7	2.5	Discovered boys bring shock, joy
-0.1	12.5	After Iraq trip, Clinton proposes war limits	0.4	16.5	Move to ban iPods from crossing the street
-0.4	50.0	Hussein's niece pleads for father's life	1.0	0.0	'Sunshine' Goydos wins Sony open
-0.4	50.5	Cheney to Congress: Can't run Iraq war by committee	0.7	2.0	Sarkozy letter surprises French cartoons hearing
-0.6	72.3	7 dead in apartment building fire	0.4	1.1	Press sees hope in Mecca talks
-0.5	59.0	Global National Major child porn ring bust	0.4	1.1	Bernhard set to leave Volkswagen
-0.6	55.4	Filipino woman kidnapped in Nigeria	0.7	8.0	India's Taj Mahal gets facelift
-0.6	50.0	Mountain glaciers melting faster, United Nations says	1.0	0.0	'Stomp' steps to No. 1 at box office

Table 7: Affective Text "Sadness" dataset, Hard (agreement <0) and Easy Cases (agreement >0.4). The headlines are from Sem2007 and the labels are from SemAnno.

Index

Wissenschaftlicher Werdegang des Verfassers[¶]

2001–2005	Studium der Musiktheorie und Sprachwissenschaft State University of New York at Buffalo
2005	Abschluss als Bachelor of Arts Bachelor-Thesis im Bereich Musik: „Summary Analysis of the Intonation in F.D.R's 'December Seventh' Speech" Referent: Prof. Martha Hyde, PhD. Bachelor-Thesis im Bereich Sprachwissenschaft: „Language and Music: Use of Musical Foundations to Explore Intonational Phonology" Referent: Prof. Michelle Gregory, PhD.
2005–2008	Studium der Sprachwissenschaft The Ohio State University
2008	Abschluss als Master of Arts Master-Thesis im Bereich Sprachtechnologie: „Automatic Coreference Resolution in Spoken Language" Referenten: Prof. Chris Brew, PhD.
seit 2011	Stipendiat am Ubiquitous Knowledge Processing Lab Technische Universität Darmstadt

Ehrenwörtliche Erklärung[‡]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades „Doktor der Naturwissenschaften" mit dem Titel *„Crowdsource Annotation and Automatic Reconstruction of Online Discussion Threads"* selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 14. December 2015

Emily K. Jamison

[¶] Gemäß § 20 Abs. 3 der Promotionsordnung der Technischen Universität Darmstadt.

[‡] Gemäß § 9 Abs. 1 der Promotionsordnung der Technischen Universität Darmstadt.

Publikationsverzeichnis des Verfassers

Emily K. Jamison and Iryna Gurevych: 'Noise or additional information? Leveraging crowdsource annotation item agreement for natural language tasks.', in: *Proceedings of EMNLP 2015: Conference on Empirical Methods in Natural Language Processing* (EMNLP 2015), Lisbon, Spain, 2015.

Emily K. Jamison and Iryna Gurevych: 'Adjacency Pair Recognition in Wikipedia Discussions using Lexical Pairs', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing* (PACLIC), Phuket, Thailand, 2014.

Emily K. Jamison and Iryna Gurevych: 'Needle in a Haystack: Reducing the Costs of Annotating Rare-Class Instances in Imbalanced Datasets', in: *Proceedings of the The 28th Pacific Asia Conference on Language, Information and Computing* (PACLIC), Phuket, Thailand, 2014.

Emily K. Jamison and Iryna Gurevych: 'Headerless, Quoteless, but not Hopeless? Using Pairwise Email Classification to Disentangle Email Threads', in: *Proceedings of 9th Conference on Recent Advances in Natural Language Processing* (RANLP 2013), Hissar, Bulgaria, 2013.

Emily K. Jamison: 'Using Grammar Rule Clusters for Semantic Relation Classification', in: *Proceedings for the ACL Workshop R̈ELMS 2011: Relational Models of Semantics̈*, Portland, Oregon, 2011.

Emily K. Jamison: 'Using Online Knowledge Sources for Semantic Noun Clustering', in: *Proceedings of the Sixth Meeting of the Midwest Computational Linguistics Colloquium* (MCLC), Bloomington, IN, USA, 2009.

Emily K. Jamison: 'CACTUS: A User-friendly Toolkit for Semantic Categorization and Clustering in the Open Domain', in: *Proceedings of the NSF Sponsored Symposium on Semantic Knowledge Discovery, Organization and Use*, New York, NY, 2008.

Emily K. Jamison and Dennis Mehay: 'OSU-2: Generating Referring Expressions with a Maximum Entropy Classifier', in: *Proceedings of the 5th International Natural Language Generation Conference* (INLG), Salt Fork, OH, USA, 2008.

Emily K. Jamison: 'Using Discourse Features for Referring Expression Generation', in: *Proceedings of the 5th Meeting of the Midwest Computational Linguistics Colloquium* (MCLC), East Lansing, MI, USA, 2008.