SUBJECTIVITY WORD SENSE DISAMBIGUATION: A TOOL FOR SENSE-AWARE SUBJECTIVITY ANALYSIS

by

Cem Akkaya

Dipl.-Ing., Technische Universität Wien, 2005

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2013

UNIVERSITY OF PITTSBURGH

INTELLIGENT SYSTEMS PROGRAM

This dissertation was presented

by

Cem Akkaya

It was defended on

June 20, 2013

and approved by

Janyce Wiebe, PhD, Professor, Department of Computer Science

Diane Litman, PhD, Professor, Department of Computer Science

Miloš Hauskrecht, PhD, Associate Professor, Department of Computer Science

Adam J. Lee, PhD, Assistant Professor, Department of Computer Science

Dissertation Director: **Janyce Wiebe, PhD**, Professor, Department of Computer Science

SUBJECTIVITY WORD SENSE DISAMBIGUATION:

A TOOL FOR SENSE-AWARE SUBJECTIVITY ANALYSIS

Cem Akkaya, PhD

University of Pittsburgh, 2013

Subjectivity lexicons have been invaluable resources in *subjectivity analysis* and their creation has been an important topic in subjectivity analysis. Many systems rely on these lexicons. For any subjectivity analysis system, which relies on a subjectivity lexicon, *subjectivity sense ambiguity* is a serious problem. Such systems will be misled by the presence of subjectivity clues used with objective senses called *false hits*.

We believe that any type of subjectivity analysis system relying on lexicons will benefit from a sense-aware approach. We think *sense-aware subjectivity analysis* has been neglected mostly because of the concerns related to *word sense disambiguation* (WSD), the problem of automatically determining which sense of a word is activated by the use of the word in a particular context according to a sense-inventory. Although WSD is the perfect tool for sense-aware classification, trust in traditional fine-grained WSD as an enabling technology is not high due to previous mostly unsuccessful results.

In this thesis, we investigate feasible and practical methods to avoid these false hits via sense-aware analysis. We define a new coarse-grained WSD task capturing the right semantic granularity specific to subjectivity analysis.

TABLE OF CONTENTS

1.0 INTRODUCTION 1

1.1 Motivation for Sense-Aware Subjectivity Analysis 2

1.2 Research Summary 3

1.3 General Hypotheses 6

1.4 Main Contributions 7

1.5 Outline 8

2.0 BACKGROUND 10

2.1 Subjectivity 10

2.1.1 MPQA Corpus and Subjectivity Lexicon 12

2.2 Word Senses 12

2.2.1 WordNet 13

2.3 Subjectivity Sense Labeling 14

2.4 Subjectivity Sense Tagging 16

3.0 SUBJECTIVITY WORD SENSE DISAMBIGUATION 18

3.1 Potential of Sense-Aware Subjectivity Analysis 19

3.2 Task Definition 20

3.3 SWSD Method 23

3.3.1 WSD features for SWSD 23

3.3.2 Subjectivity features for SWSD . . . 23
3.3.3 Training . . . 24
3.4 Experimental Design . . . 25
3.4.1 Data Creation . . . 26
3.4.2 In Vivo Evaluation . . . 27
3.4.2.1 Results on Coarse-grained Training . . . 27
3.4.2.2 Results on Fine-grained Training . . . 29
3.4.2.3 Extending SWSD with Subjectivity Features . . . 29
3.4.3 In Vitro Evaluation . . . 30
3.4.3.1 MPQA Coverage . . . 30
3.4.3.2 Rule-based Classifier . . . 32
3.4.3.3 Contextual Subjective/Objective Classifier . . . 34
3.4.3.4 Contextual Polarity Classifier . . . 36
3.5 Summary and Discussion . . . 40
3.6 Related Work . . . 42
4.0 NON-EXPERT ANNOTATIONS . . . 46
4.1 Amazon Mechanical Turk . . . 47
4.2 Amazon Mechanical Turk for SWSD . . . 49
4.2.1 Subjectivity Sense Tagging via Amazon Mechanical Turk . . . 49
4.2.2 Annotation Quality . . . 51
4.2.2.1 Experimental Design . . . 51
4.2.2.2 Group Evaluation . . . 55
4.2.2.3 Worker Evaluation . . . 56
4.2.2.4 Learning Effect . . . 57
4.3 SWSD on Non-expert Annotations . . . 58

4.3.1 In Vivo Evaluation . . . 58
4.3.2 In Vitro Evaluation . . . 60
4.3.2.1 Data Annotation . . . 60
4.3.2.2 Rule-Based SWSD Integration . . . 62
4.3.2.3 Learning SWSD Integration . . . 64
4.4 Summary and Discussion . . . 67
4.5 Related Work . . . 70
5.0 REDUCING ANNOTATION EFFORT: CLUSTER AND LABEL . . . 72
5.1 Context Clustering . . . 73
5.1.1 Distributional Semantic Models . . . 73
5.2 Compositional Models . . . 75
5.2.1 Exploiting Richer Contexts . . . 77
5.2.2 Experiments . . . 80
5.2.2.1 Semantic Space . . . 80
5.2.2.2 Context Representations . . . 81
5.2.2.3 Clustering Algorithm and Evaluation Metric . . . 82
5.2.2.4 Effect of Longer Dependencies and Filtering Strategies . . . 83
5.2.2.5 Comparison of Context Representations . . . 85
5.2.2.6 Merging Context Representations . . . 85
5.2.3 Incorporating Subjectivity into DSMs . . . 87
5.3 Labeling Clusters . . . 90
5.3.1 Constrained Clustering . . . 91
5.3.2 Iterative Constrained Clustering . . . 92
5.3.2.1 Informativeness . . . 92
5.3.2.2 Imposing Constraints . . . 94

5.3.2.3 Complete Algorithm 95
5.3.3 Experiments 96
5.3.3.1 Compared Methods 96
5.3.3.2 Effect of Active Selection Strategy 98
5.3.3.3 Effect of Metric Learning 101
5.3.3.4 Effect of Oracle Cluster Assignment 101
5.3.3.5 SWSD on semi-automatically generated annotations 104
5.4 Summary and Discussion 108
5.5 Related Work 109
6.0 CONCLUSIONS AND FUTURE DIRECTIONS 113
BIBLIOGRAPHY 119

LIST OF TABLES

1 Results of SWSD with coarse-grained training on senSWSD. 28

2 Results of SWSD with fine-grained training on senSWSD. 29

3 Results of SWSD: fine-grained training vs. coarse-grained training 30

4 Results of SWSD: effect of subjectivity features on SWSD 31

5 Effect of SWSD on the Rule-based Classifiers. 32

6 Effect of SWSD on the Subjective/Objective Classifier. 36

7 Effect of SWSD on the Neutral/Polar Classifier. 38

8 Effect of SWSD on the Contextual Polarity Classifier. 39

9 Frequent label percentages of the target words in the MTurk experiment. . . 52

10 Constraints for each HIT group. 53

11 Accuracy and kappa scores for each group of workers. 56

12 Spammer representation in groups. 57

13 Comparison of SWSD systems 59

14 S/O classifier with and without SWSD. 63

15 N/P classifier with and without SWSD 64

16 S/O classifier with learned SWSD integration 65

17 N/P classifier with learned SWSD integration 66

18 Polarity classifier with and without SWSD. 67

19 A hypothetical word-word co-occurrence matrix 74
20 Additive and multiplicative composition of co-occurrence vectors 76
21 Effect of the various dependency path lengths and filtering techniques used to compute the contextual representation on the clustering performance 84
22 Comparison of context representations for context clustering on SENSEVAL . 85
23 Comparison of context representations for context clustering on senSWSD . . 86
24 Effect of merging context representations 87
25 Comparison of context representations for context clustering on SENSEVAL on sample words 88
26 Effect of DSM modification 89
27 Annotation Reduction with ICC over Uncertainty and Random Sampling . . 106
28 S/O classifier with SWSD trained on semi-automatically generated annotations 107
29 N/P classifier with SWSD trained on semi-automatically generated annotations 108

LIST OF FIGURES

1 WordNet senses for the noun “alarm” 14

2 Subjectivity sense labels – subjective examples 15

3 Subjectivity sense labels – objective examples 15

4 Sense-aware subjectivity analysis relying on WSD. 19

5 WordNet senses for the noun “pain” 21

6 Sense-aware subjectivity analysis relying on SWSD. 22

7 WSD features for SWSD 24

8 Subjectivity features for SWSD 25

9 SWSD integration to contextual subjectivity classifier. 35

10 SWSD integration to contextual polarity classifier. 37

11 Sense sets for target word “appear”. 50

12 Venn diagram illustrating worker distribution. 55

13 Example for compositional representation 77

14 Example for distributional substitutes 79

15 WSD features for SWSD 82

16 Behaviour of selection function 94

17 Accuracy of generated subjectivity sense tagged data – ICC vs. random selection 99

18 Accuracy of generated subjectivity sense tagged data – ICC without soft-contraints vs. Klein 100
19 Accuracy of semi-automatically created data by ICC with and without soft-constraints 102
20 Accuracy of semi-automatically created data by ICC with oracle cluster assignment 103
21 Accuracy of semi-automatically created data by ICC and baselines 105

LIST OF ALGORITHMS

1 Iterative Constrained Clustering 96

1.0 INTRODUCTION

Subjectivity Analysis [Wiebe et al., 1999, Wiebe et al., 2004] is the automatic extraction of linguistic expressions of private states in text. A private state is defined by [Quirk et al., 1985] as a state that is not open to objective observation or verification. They are mental and emotional states such as opinions, beliefs, sentiments, emotions, goals, evaluations, stances, and speculations.

Subjectivity analysis is an active area of research in *Natural Language Processing* (NLP). It is largely motivated by the need to automatically analyse opinions and emotions in text to support NLP applications. The advance of the *World Wide Web* and *Social Media*, which made vast amount of opinionated user content available, is one of the driving forces behind this research field.

Many approaches to subjectivity analysis rely on lexicons of words that may be used to express subjectivity. We call these lexicon entries subjectivity clues. Examples of such clues from an established subjectivity lexicon [Wiebe et al., 2005b, Wilson, 2007] are the following (in bold):

(1.1) He is a **disease** to every team he has gone to.

(1.2) Converting to SMF is a **headache**.

(1.3) The concert left me **cold**.

(1.4) That guy is such a **pain**.

Knowing the subjectivity and the semantic orientation (i.e. polarity) of these clues would help a system recognize the negative sentiments in these sentences. Thus, subjectivity analysis systems typically look for the presence of these clues in text. They may rely only on this information, or they may combine it with additional information (e.g. discourse relations) as well.

1.1 MOTIVATION FOR SENSE-AWARE SUBJECTIVITY ANALYSIS

Subjectivity lexicons have been invaluable resources in subjectivity analysis and their creation has been an important topic in subjectivity analysis [Hatzivassiloglou and McKeown, 1997, Turney, 2002a, Gamon and Aue, 2005, Esuli and Sebastiani, 2006a]. However, even manually-developed subjectivity lexicons have significant degrees of subjectivity sense ambiguity. This means many entries in the lexicon have both subjective and objective senses. False hits – subjectivity clues used with objective senses – are a significant source of error in subjectivity analysis. A study on the MPQA opinion-annotated corpus [Akkaya et al., 2009] shows that 42.9% of the clue instances in the MPQA Corpus are false hits. This demonstrates how serious the ambiguity is. For example, even though the following sentence contains all of the negative keywords from the examples above, it is nevertheless objective, as the keywords are all used with objective senses:

(1.5) Early symptoms of the **disease** include severe **headaches**, red eyes, fevers and **cold** chills, body **pain**, and vomiting.

A subjectivity analysis system relying on a subjectivity lexicon will be misled by the presence of such false hits. A possible solution to this problem is creating subjectivity lexicons listing word senses instead of simple keywords and doing sense-aware subjectivity analysis,

where clue instances in text are disambiguated for their senses to avoid false hits. There have been recent efforts to create subjectivity lexicons listing word senses [Andreevskaia and Bergler, 2006, Wiebe and Mihalcea, 2006, Su and Markert, 2009]. Primarily, subjectivity lexicons have only been applied as conventional subjectivity lexicons by aggregating sense level information to the word level. We believe that any type of subjectivity analysis system relying on lexicons will benefit from a sense-aware approach.

We think sense-aware subjectivity analysis has been neglected mostly because of the concerns related to *Word Sense Disambiguation* (WSD), the problem of automatically determining which sense of a word is activated by the use of the word in a particular context according to a sense-inventory. Although WSD is the perfect tool for sense-aware classification, trust in traditional fine-grained WSD as an enabling technology is not high due to previous mostly unsuccessful results in applications such as *Information Retrieval* and *Document Classification*.

1.2 RESEARCH SUMMARY

There are three major obstacles for utilizing sense information in subjectivity analysis. First of all, the sense inventories utilized in WSD are very fine-grained. Even the best performing supervised WSD systems are not accurate. That is not surprising considering that even for trained humans it is hard to distinguish between fine-grained senses of a word. Utilizing such a noisy information for sense-aware analysis will not be optimal. WSD systems disambiguate each target word separately and each word needs separate training data, but sense-tagged corpora to train WSD systems are very limited in availability and hard to create manually. This brings us to the second obstacle, namely the knowledge acquisition bottleneck. The third obstacle is the sparsity problem that will form by utilizing fine-grained senses of

subjectivity clues. In our research, we aim to target these obstacles in order to accomplish successful sense-aware subjectivity analysis.

We think that fine sense distinctions, which make WSD a hard task, are not required for sense-aware subjectivity analysis. We do not need to pinpoint the exact sense of a word in context, we just need to know if the word is used with a subjective sense or an objective sense. Following this insight, we define a coarse-grained WSD task, *Subjectivity Word Sense Disambiguation* (SWSD), which disambiguates two senses of a word: (1) a subjective sense and (2) an objective sense. SWSD aims to capture the right semantic granularity specific to subjectivity analysis. There are two high level goals we want to accomplish :

- **Goal 1**: *We want to show that SWSD can provide reliable sense subjectivity information and that we can utilize it to improve contextual subjectivity. We target accuracy and sparsity issues to accomplish this goal.*
- **Goal 2**: *We want to show that it is feasible to obtain large amounts of training data for SWSD rendering it a practical technology. We target knowledge acquisition bottleneck to accomplish this goal.*

For our first goal, we build a SWSD system to disambiguate instances of subjectivity clues as being used with a subjective sense or an objective sense. In this phase, we utilize sense-tagged data produced by human experts to train our system, which is very limited. Thus, the impact of SWSD is also limited by the number of the clues for which we have sense-tagged data. Since our first goal is to prove the feasibility of SWSD and show its impact on contextual subjectivity analysis, the limited amount of data is not a big concern. We are interested in the applicability of SWSD as an enabling technology. Since we conceptualize SWSD as a tool to achieve sense-aware subjectivity analysis, we want to see improvements in end systems. We work on methods to integrate SWSD into contextual subjectivity analysis

systems. We conduct two types of evaluation: in vivo and in vitro. In vivo evaluation of SWSD gives us an idea about its performance as a standalone task. In vitro evaluation of SWSD shows us if SWSD can have a positive impact on contextual subjectivity analysis via sense-aware classification.

For our second goal, we try to reduce annotation time and cost to obtain training data for SWSD. Although the definition of SWSD allows easier data annotation – choosing between two senses vs. choosing from a fine-grained list senses –, we implement additional steps to take on the problem of the knowledge acquisition bottleneck. We follow two very different approaches.

Our first approach is utilizing crowdsourcing – *Amazon Mechanical Turk* (MTurk) – in order to reduce annotation time and cost. We can obtain large amounts of non-expert annotations as a cheap and fast alternative to expert annotations. We face multiple challenges due to the unique properties of the MTurk environment as an annotation source and also due to the definition of our task. The annotations obtained via MTurk are noisy by nature, since the workers are not trained on the underlying annotation task and some of them are just spammers. We need to deal with these challenges to obtain reliable non-expert SWSD annotations. For this purpose, we propose a simple representation of the annotation task suitable for the MTurk environment and investigate applicability of built-in control mechanisms in the MTurk environment as a filter for spammers. Again we conduct in vivo and in vitro evaluation of SWSD trained on non-expert annotations. As we experiment with this approach, we explore the following general hypotheses:

Our second approach is semi-automatically generating annotated data. We explore the application of a "cluster and label" strategy. Basically, we cluster unlabeled word instances into coherent clusters – in terms of the meanings the word instances have – and then label clusters as a whole instead of labelling all the instances of a word separately. Of course, such

an approach introduces noise in the labeled data that we want to keep minimal. Thus, we experiment with novel techniques to obtain an expressive representation of word meaning. In addition, we work on a novel semi-supervised clustering algorithm to incorporate some prior knowledge into the clustering process and minimize noise as much as possible. Again we conduct in vivo and in vitro evaluation of SWSD trained on semi-automatically generated annotations.

To summarize, sense-aware subjectivity analysis is a neglected line of research representing real problems. It has the promise to improve any type of subjectivity analysis system relying on subjectivity lexicons. In our research, we fill in this missing piece.

1.3 GENERAL HYPOTHESES

In our research, we follow as step-by-step approach targeting problems associated with sense-aware subjectivity analysis. As we work towards our goals, we explore following general hypotheses addressed in different chapters throughout this dissertation.

- **Hypothesis 1**: S/O sense groupings are natural and both groups can be disambiguated accurately by a supervised model.
- **Hypothesis 2**: The subjectivity sense information provided by SWSD is more reliable than the fine-grained sense information provided by WSD.
- **Hypothesis 3**: SWSD can be exploited to improve the performance of contextual subjectivity analysis systems via sense-aware analysis.
- **Hypothesis 4**: Crowdsourcing can be utilized to collect high-quality SWSD annotations in order to train SWSD classifiers with a good performance.

- **Hypothesis 5**: A "cluster and label" strategy together with some prior knowledge can be utilized to reduce annotation effort to train reliable SWSD classifiers.

1.4 MAIN CONTRIBUTIONS

The research in this dissertation contributes to an on-going line of research in subjectivity analysis and lexical disambiguation. The main contribution of this thesis is to establish sense information as a useful information for contextual subjectivity analysis.

We show that SWSD, as a coarse-grained WSD task, can be done reliably and can improve contextual subjectivity analysis via sense-aware classification. We are the first ones to conceptualize the task SWSD and use it for sense-aware subjectivity classification. The findings are also important for the lexical disambiguation community because of the previous mixed results for WSD as an enabling technology. Our research is a representative of application-specific WSD, which is considered a promising next step in WSD [Agirre and Edmonds, 2006].

This research provides general strategies to integrate SWSD information into contextual subjectivity analysis. The integration of SWSD to the underlying subjectivity analysis system is important. How we do the integration heavily depends on the properties of the underlying system. We define strategies for various subjectivity analysis systems.

We show that a simple representation of the annotation task suitable for the MTurk environment allows us to collect reliable non-expert annotations. We demonstrate that non-expert SWSD can be utilized to improve contextual subjectivity analysis.

We rely on a "cluster and label" strategy to reduce annotation effort. To achieve best possible purity, we experiment with novel methods for context representation and semi-supervised clustering. We propose a new semi-supervised clustering algorithm with active

constraint selection. We see that our active selection improves over previous work. Ultimately, we show that we can reduce the annotation effort by 41% without any loss in performance. The proposed method is not limited to SWSD. It is also applicable for the general WSD task.

As part of the "cluster and label" approach, we define a novel model for representing meaning in context, which extends on an existing method for compositional semantics. When we utilize this representation for context clustering, we achieve significant improvement over previous approaches. These results have implications for various lexical disambiguation tasks such as word sense discrimination, paraphrase recognition, and textual entailment.

1.5 OUTLINE

Chapter 2 provides the background knowledge on linguistic subjectivity, word senses and their relation. In this chapter, we also introduce two important annotation tasks on which we rely for our research. The remainder of the thesis follows a straightforward structure. Each hypothesis from section 1.3 is explored in order. In Chapter 3, we define our task SWSD and present our experiments on feasibility and applicability of SWSD. This chapter deals with our first goal and explores the first three hypotheses. Part of the research presented in this chapter is published in [Akkaya et al., 2009]. Chapter 4 summarizes our work on crowdsourcing. The focus of this chapter is SWSD relying on non-expert annotations. The chapter deals with our second goal and explores the fourth hypothesis. The research presented in this chapter is published in [Akkaya et al., 2010, Akkaya et al., 2011]. In Chapter 5, we present our work on reducing annotion effort via a "cluster and label" strategy. We propose novel methods to improve quality of semi-automatically labeled data for SWSD. The chapter deals with our second goal and explores the fifth hypothesis. Part of the research

presented in this chapter is published in [Akkaya et al., 2012]. Finally, in Chapter 6, we summarize the thesis contributions and discuss future directions.

2.0 BACKGROUND

In this chapter, we aim to introduce two major concepts vital for our research: *subjectivity* and *word senses*. In Section 2.1, we will introduce the concept of subjectivity and related resources made use of in this research. Then, we will look at the concept of word senses in Section 2.2. In Section 2.3 and 2.4, we will discuss two annotation tasks related to the subjectivity of word senses.

2.1 SUBJECTIVITY

We adopt the definitions of *subjective* and *objective* from [Wiebe et al., 2005b, Wilson, 2007, Wiebe and Mihalcea, 2006]. Subjective expressions are words and phrases being used to express mental and emotional states, such as speculations, evaluations, feelings, emotions, stances and beliefs. A general covering term for such states is *private state* [Quirk et al., 1985], an internal state that cannot be directly observed or verified by others. [Wiebe and Mihalcea, 2006] give the following examples:

(2.1) His **alarm** grew.

(2.2) He **absorbed** the information quickly.

(2.3) UCC/Disciples leaders **roundly condemned** the Iranian President's

(2.4) **verbal assault** on Israel.

(2.5) **What's the catch?**

Subjective/objective (S/O) distinction allows us to discriminate between objective and subjective content. For example, such a discrimination is beneficial for question answering [Stoyanov et al., 2005] and information extraction [Riloff et al., 2005]. Feeding a system only with factual data or only with non-factual data depending on the system's needs yields better performance. Polarity (also called *semantic orientation*) is also important to NLP applications. For example, in product review mining, we want to know whether an opinion about a product is positive or negative.

The contextual subjectivity analysis experiments in this work include both S/O and polarity classifications. Note that some other researchers generally associate polarity classification with sentiment analysis. In our group, we consider sentiment as a specific subtype of subjectivity, namely linguistic expressions of evaluations, stances and emotions. Thus, we use subjectivity as a general term.

It is important to point out the relation between polarity and subjectivity for later sections. An objective expression – an expression which is not subjective – does not show any semantic orientation. On the contrary, a subjective expression does not necessarily have to show any particular semantic orientation. Although it is more probable that a subjective expression has a *positive* or *negative* semantic orientation, its semantic orientation may also be neutral. In the examples below, while the subjective expression in the first snippet does not have any semantic orientation (*neutral*), the subjective expression in the second snippet has a positive semantic orientation.

(2.6) Certainly, the intensive efforts that **our big sister**, Egypt, made ...

(2.7) This will be **a big deal** down there ...

2.1.1 MPQA Corpus and Subjectivity Lexicon

The Multi Perspective Question Answering (MPQA) Opinion Corpus [1] [Wiebe et al., 2005b, Wilson, 2007], is annotated for subjective expressions of varying lengths from single words to long phrases. It consists of 535 documents from 187 different new resources. The annotations are done according to the MPQA subjectivity annotation scheme [Wiebe et al., 2005a,Wiebe, 2002, Wiebe, 1994]. The annotations hold three major components: (1) the *source* of the private state, (2) the expression of the private state, (3) the *target* of the private state. The annotation scheme differentiates between various subjectivity types such as *sentiment, arguing, agreement*, and *speculation*. In addition, properties of these private states are also annotated including *polarity* and *intensity*.

Another important resource we make us of is the subjectivity lexicon[2] introduced in [Wilson et al., 2005], which contains approximately 8000 words which may be used to express subjectivity. Each entry consists of a subjective word, its prior polarity (*positive/negative/neutral*), morphological information, and part of speech information. Moreover, the words are grouped according to their reliability as subjectivity clues. Words that are subjective in most contexts are marked as strongly subjective (*strongsubj*), and those that may only have certain subjective usages are marked weakly subjective (*weaksubj*).

2.2 WORD SENSES

In this section, we will not go into detail of philosophical accounts of meaning (e.g. *Gricean* and *Fregean* models). We will concentrate on the more practical view hold by lexicographers. The meaning of a word is context sensitive. Different meanings of a word may be related to

[1] Available at http://mpqa.cs.pitt.edu/corpora
[2] Available at http://mpqa.cs.pitt.edu/lexicons

each other, in which case we speak from polysemy. They may also be completely unrelated, in which case we speak from homonymy. For example, the "financial institution" meaning and the "river side" meaning are two unrelated senses of the noun "bank". In comparison, when we talk about a "financial institution", we may refer to a company or to an actual building, which are two related senses of the noun "bank". The senses of words can be found in dictionaries, which are created on evidence found in text corpora by lexicographers. Lexicographers use concordance information in large text corpora and mine frequent usage patterns in order to explore and describe different senses of a word. [Kilgarriff, 1997] describes this process in detail. A lexicographer collects occurrences of a word in a corpus and tries to cluster different usages into coherent sets. All the instances in a set should have more in common with the other instances of that set, than with any instance of any other set. Then, it is the lexicographer's job to describe the common attributes and semantic content in a cluster and code this information as a dictionary definition. The target audience of a dictionary effects how usages are clustered by a lexicographer and also which clusters are lexicalized as a dictionary definition.

2.2.1 WordNet

Word Sense Disambiguation research is driven by the need for a sense inventory (i.e. dictionary). WordNet [Miller, 1995] provides such a sense inventory. WordNet lists different senses of words and groups them into sets of synonyms called synsets. It also provides semantic relations between synsets. In Figure 1, we see the synsets for the noun "alarm" extracted from WordNet. For each target synset, we also print the corresponding hypernym (i.e. parent in a kind-of relation).

Generally, WordNet is too fine-grained for most applications. Because of that, unneces-

{alarm, dismay, consternation} – fear resulting from the awareness of danger => {fear, fearfulness, fright} – an emotion experienced in anticipation of some specific pain or danger (usually accompanied by a desire to flee or fight)
{alarm, warning device, alarm system} – a device that signals the occurrence of some undesirable event => {device} – an instrumentality invented for a particular purpose; "the device is small enough to wear on your wrist"; "a device intended to conserve water"
{alarm, alert, warning signal, alarum} – an automatic signal (usually a sound) warning of danger => {signal, signaling, sign} – any nonverbal action or gesture that encodes a message; "signals from the boat suddenly stopped"
{alarm clock, alarm} – a clock that wakes a sleeper at some preset time => {clock} – a timepiece that shows the time of day

Figure 1: WordNet senses for the noun "alarm"

sary errors may occur. That is not surprising considering that even for a human it is hard to distinguish between fine-grained senses of a word hitting a ceiling of 80% inter-annotator agreement [Edmonds and Kilgarriff, 2002]. Nevertheless, WordNet has been the choice of the sense inventory for WSD due to its availability and coverage.

2.3 SUBJECTIVITY SENSE LABELING

For SWSD, we need the notions of subjective and objective *senses* of words in a dictionary. We adopt the definitions from [Wiebe and Mihalcea, 2006], who describe their annotation scheme as follows.

Classifying a sense as S means that, when the sense is used in a text or conversation, one expects it to express subjectivity, and also that the phrase or sentence containing it to expresses subjectivity. Figure 2 holds subjective examples given in [Wiebe and Mihalcea,

2006].

His **alarm** grew.
{alarm, dismay, consternation} – fear resulting from the awareness of danger
=> {fear, fearfulness, fright} – an emotion experienced in anticipation of some specific pain or danger (usually accompanied by a desire to flee or fight)
What's the **catch**?
{catch} – a hidden drawback; "it sounds good but what's the catch?"
=> {drawback} – the quality of being a hindrance; "he pointed out all the drawbacks to my plan"

Figure 2: Subjectivity sense labels – subjective examples

Classifying a sense as *O* means that, when the sense is used in a text or conversation, one does not expect it to express subjectivity. Figure 3 holds objective examples given in [Wiebe and Mihalcea, 2006].

The **alarm** went off.
{alarm, warning device, alarm system} – a device that signals the occurrence of some undesirable event
=> {device} – an instrumentality invented for a particular purpose; "the device is small enough to wear on your wrist"; "a device intended to conserve water"
He sold his **catch** at the market.
{catch, haul} – the quantity that was caught; "the catch was only 10 fish"
=> {indefinite quantity} – an estimated quantity

Figure 3: Subjectivity sense labels – objective examples

They also note that a phrase or a sentence containing an objective sense does not necessarily need to be objective. If the phrase or sentence containing the objective sense is subjective, the subjectivity is due to something else. Consider the following examples from [Wiebe and Mihalcea, 2006]:

(2.8) Will someone shut that damn **alarm** off?

(2.9) Can't you even **boil** water?

While these sentences contain objective senses of alarm and boil, the sentences are subjective nonetheless. But they are not subjective due to alarm and boil, but rather to punctuation, sentence forms, and other words in the sentence. Finally, classifying a sense as B means it covers both subjective and objective usages.

[Wiebe and Mihalcea, 2006] performed an agreement study and report that a good agreement (κ=0.74) can be achieved between human annotators labelling the subjectivity of senses. For a similar task, [Su and Markert, 2008] also report a good agreement (κ=0.79).

Note that subjectivity sense labelling is different from subjectivity sense tagging, where one annotates a word instance in text as being used with an objective or a subjective sense.

2.4 SUBJECTIVITY SENSE TAGGING

The training and test data for SWSD consists of word instances in a corpus labeled as S or O, indicating whether they are used with a subjective or objective sense. *Subjectivity sense tagging* refers to the annotation of word instances in text. The same definition of subjective and objective sense from section 2.3 is used for subjectivity sense tagging.

In the examples below, first instance of attack should be tagged as S , because it is used with a subjective sense. Second instance is used with an objective sense. Thus, it should be tagged as O.

(2.10) Ivkovic had been a target of intra-party feuding that has shaken the party. He was **attacked** by Milosevic for attempting to carve out a new party from the Socialists.

(2.11) A new treatment based on training T-cells to **attack** cancerous cells is being developed at the University of Pennsylvania.

In this research, subjectivity sense tagging was done according to subjectivity sense labeled sense inventories. This means subjectivity sense labeling was a prior step for subjectivity sense tagging. As mentioned earlier, WordNet is not built with subjectivity in mind – this is also true for other dictionaries. Thus, it misses some subjective and objective meanings and even mixes them together into the same synset. To handle this problem, we also conduct subjectivity sense tagging according to usage inventories. Usage inventories are basically sets of sample subjective and objective usages of a word extracted from text. We will discuss usage inventories in more detail in Section 4.3.2.1.

3.0 SUBJECTIVITY WORD SENSE DISAMBIGUATION

For any subjectivity analysis system, which relies on a subjectivity lexicon, subjectivity sense ambiguity is a serious problem. Such systems will be misled by the presence of false hits – subjectivity clues used with objective senses. *Sense-aware subjectivity analysis* is a solution to avoid false hits and errors. A straightforward way to achieve sense-aware treatment is to provide the sense of each clue instance to the subjectivity analysis system. The sense information can be obtained via *Word Sense Disambiguation* (WSD). Figure 4 illustrates this straightforward scenario.

There are three major problems with this approach. First of all, fine-grained sense information provided is not very reliable and will introduce noise. Supervised WSD is a must even for a moderately accurate performance and the annotation effort to create fine-grained sense tagged training data is very time-consuming and expensive. Thus, knowledge acquisition bottleneck is another problem. Third, the input of fine-grained senses might result in sparsity, because it will increase the number of features input to the underlying subjectivity analysis system. Each clue will result in as many features as the number of the senses it has. [Ar et al., 2011] shows supporting evidence. On a document-level polarity classification task, they report that sense-level information only helps if it is provided by an oracle but not if it is provided by automatic WSD.

In this chapter, we introduce a new task *Subjectivity Word Sense Disambiguation* (SWSD) targeting mentioned three problems in the way of sense-aware analysis. Our aim is to exploit

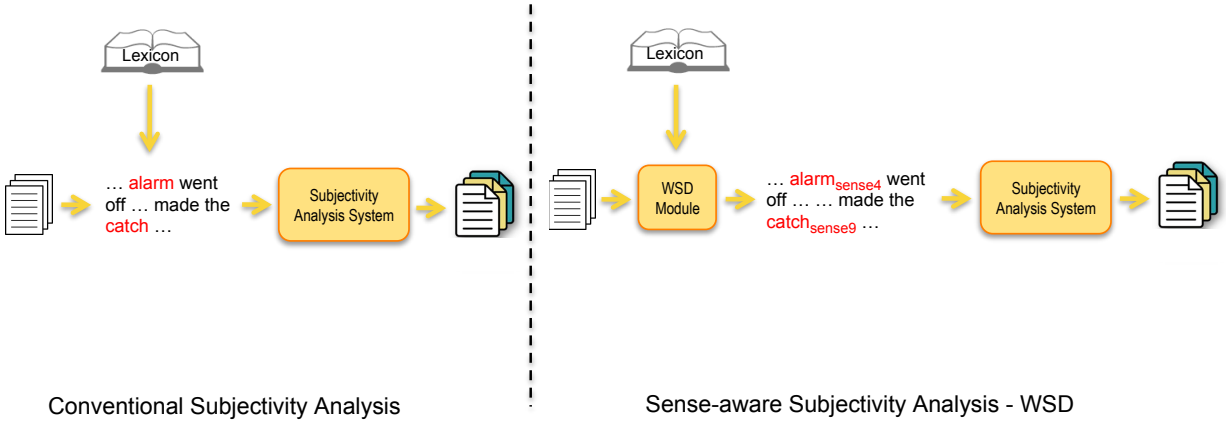

Figure 4: Sense-aware subjectivity analysis relying on WSD.

SWSD improve the performance of subjectivity analysis systems. We implement SWSD relying on supervised learning methods and integrate it to contextual subjectivity analysis. In this chapter, we entirely make use of expert annotations to train our SWSD models.

In Section 3.1, we represent statistics to demonstrate the potential benefit of performing SWSD. Section 3.2 describes how we conceptualize SWSD. Section 3.3 provides implementation details. In Section 3.4, we evaluate SWSD as a standalone task and also its impact on the contextual subjectivity analysis. The research presented in this chapter is published in [Akkaya et al., 2009].

3.1 POTENTIAL OF SENSE-AWARE SUBJECTIVITY ANALYSIS

We investigate the distribution of the clues from the subjectivity lexicon in the MPQA Corpus to show the promise of sense-aware subjectivity analysis. In our studies, we find

out that the subjectivity lexicon covers a substantial subset of the subjective expressions in the MPQA Corpus: 67.1% of the subjective expressions contain one or more lexicon entries. On the other hand, 42.9% of the instances of the lexicon entries in the MPQA Corpus are not in subjective expressions. An instance that is not in a subjective expression is, by definition, being used with an objective sense. Thus, these instances are false hits of subjectivity clues. As mentioned before, the entries in the lexicon have been pre-classified as either more (*strongsubj*) or less (*weaksubj*) reliable. We see this difference reflected in their degree of ambiguity – 53% of the *weaksubj* instances are false hits, while only 22% of the *strongsubj* instances are.

Another related finding is reported in [Wilson, 2007]. [Wilson, 2007] reports that the out-of-context lexicon polarity of a subjectivity clue (i.e. prior polarity) does not agree with the polarity of its context 48% of the time. Of these mismatches, 76% result from clues with non-neutral prior polarity appearing in phrases with neutral contextual polarity. Most probably, these clues are used with an objective sense.

To summarize, the high coverage of the lexicon in the MPQA corpus demonstrates its potential usefulness for subjectivity analysis systems, while its degree of ambiguity, in the form of false hits, shows the potential benefit of performing sense-aware subjectivity analysis.

3.2 TASK DEFINITION

In this work, we define SWSD as automatically determining the sense type - subjective sense or objective sense - of a target word in context. SWSD is as an application-specific WSD task.

{pain, hurting} – a symptom of some physical hurt or disorder; "the patient developed severe pain and distension"
=> {fear, fearfulness, fright} – an emotion experienced in anticipation of some specific pain or danger (usually accompanied by a desire to flee or fight)
{pain, painfulness} – emotional distress; a fundamental feeling that people try to avoid; "the pain of loneliness"
=> {fear, fearfulness, fright} – an emotion experienced in anticipation of some specific pain or danger (usually accompanied by a desire to flee or fight)
{pain, pain sensation, painful sensation} – a somatic sensation of acute discomfort; "as the intensity increased the sensation changed from tickle to pain"
=> {fear, fearfulness, fright} – an emotion experienced in anticipation of some specific pain or danger (usually accompanied by a desire to flee or fight)
{pain, pain in the neck, nuisance} – a bothersome annoying person; "that kid is a terrible pain"
=> {fear, fearfulness, fright} – an emotion experienced in anticipation of some specific pain or danger (usually accompanied by a desire to flee or fight)

Figure 5: WordNet senses for the noun "pain"

Consider senses of the word "pain" from WordNet in Figure 5. First and third entries are objective usages of the word pain and remaining ones are subjective usages. The objective senses are more similar to each other than they are to subjective senses and vice versa. For sense-aware subjectivity analysis we do not need to pinpoint the exact sense of the word pain. It is enough to know if the instance is used with a subjective or with an objective sense. It is a binary task. That makes our task easier than traditional fine-grained WSD. Thus, we believe that SWSD can be done with a high accuracy and that we can avoid confusion and probable errors caused by making unnecessary fine distinctions. Figure 6 illustrates the sense-aware analysis using SWSD. As we see, SWSD provides to the subjectivity analysis system the information if the clue instance has a subjective or an objective sense. Thus, sparsity is not a problem in contrast to providing specific sense information about each clue instance. In addition, the annotation task – subjectivity sense-tagging – to create training

data is easier than doing fine-grained sense-tagging.

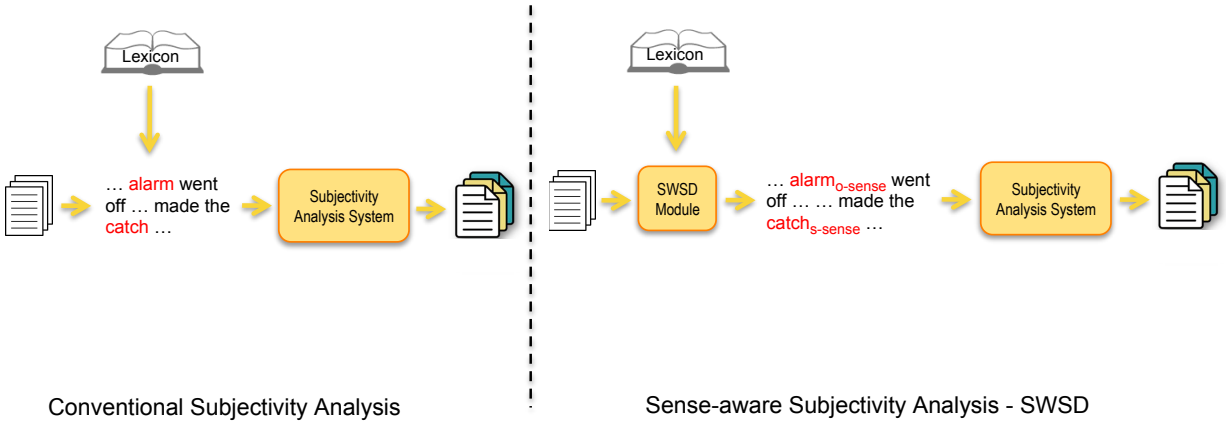

Figure 6: Sense-aware subjectivity analysis relying on SWSD.

Note that SWSD is midway between pure dictionary classification and pure contextual interpretation. For SWSD, the context of the word is considered in order to *perform* the task, but the subjectivity is determined solely by the dictionary. In contrast, full contextual interpretation can deviate from a sense's subjectivity label in the dictionary. As noted above, words used with objective senses may appear in subjective expressions. For example, an SWSD system would label the following examples of alarm as *S*, *O* and *O*, respectively. On the other hand, a sentence-level subjectivity classifier would label the sentences as *S*, *S*, and *O*, respectively.

(3.1) His **alarm** grew.
Will someone shut that darn **alarm** off?
The **alarm** went off.

3.3 SWSD METHOD

We rely on supervised learning to conduct SWSD. For each target clue, we train a different classifier. The method is similar to targeted WSD where a different classifier is trained for each target word. In contrast, all-words WSD relies on a single classifier to classify all the words in a text piece. We choose to follow a supervised and targeted method, since it performs best for WSD and we expect the same for SWSD.

3.3.1 WSD features for SWSD

We borrow machine learning features which have been successfully used in WSD research. These features try to capture information about the context of the target word instance to be disambiguated. They are grouped usually into local and topical features. Local features describe the local context of the target word instance. They capture collocations, argument-head relations and syntactic cues. On the other hand, topical features describe a larger context. They capture topic or domain of the text piece. We make use of both feature types. Specifically, we use the features listed in Figure 7 from [Mihalcea, 2002b].

3.3.2 Subjectivity features for SWSD

We extend wsd features with features aiming to capture subjectivity of the surrounding context. Some of these features are proven to be effective in contextual subjectivity analysis [Wiebe and Riloff, 2005, Wilson et al., 2005].

Sentence subjectivity features listed in Figure 8 try to capture the subjectivity of a larger context around the target word. They are comparable to the global context WSD features in terms of their scope. This set includes as features counts of strongsubj (highly reliable) and

CW : the target word itself : nominal {1}
CP : the part of speech of the target word : nominal {1}
CF : the surrounding context of 3 words and their POS : nominal {12}
HNP : the head of the noun phrase to which the target word belongs : nominal {1}
NB : the first noun before the target word : nominal {1}
VB : the first verb before the target word : nominal {1}
NA : the first noun after the target word : nominal {1}
VB : the first verb before the target word : nominal {1}
VA : the first verb after the target word : nominal {1}
SK : at most 10 context words occurring at least 5 times (for each sense) : numeric $\{<= 10*\#sense\}$

Figure 7: WSD features for SWSD

weaksubj (less reliable) clues in the sentence the target word instance appears (i.e. current), and also in the previous and the next sentence. The set also includes counts of adjectives and adverbs in the current sentence. Local subjectivity features listed 8 aim to capture the subjectivity of the immediate neighbours of the target word. Thus, they are comparable to the local context WSD features. They capture the relation to other subjectivity clues.

3.3.3 Training

As mentioned before, SWSD automatically determines which sense type (i.e. S or O) is activated in a specific context. We have two options to accomplish this. We can train our SWSD classifiers on subjectivity sense-tagged data 2.4 in a straightforward fashion. We refer to this approach as coarse-grained training. Another approach, though not optimal, is to train a classifier on the fine-grained sense tagged data. Then the system can collapse the fine-grained sense distinctions, output by the classifier, to sense types (i.e. S or O) according to a subjectivity sense labeled sense inventory 2.3. We refer to this approach as fine-grained

Sentence subjectivity features:
SC : number of strongly subjective clues in the current sentence : numeric {1}
SP : number of strongly subjective clues in the previous sentence : numeric {1}
SN : number of strongly subjective clues in the next sentence : numeric {1}
WC : number of weakly subjective clues in the current sentence : numeric {1}
WP : number of weakly subjective clues in the previous sentence : numeric {1}
WN : number of weakly subjective clues in the next sentence : numeric {1}
CB : number of adjectives in the current sentence : numeric {1}
CB : number of adverbs in the current sentence : numeric {1}

Local subjectivity features:
MSC : the membership of the surrounding context of 3 words to the subjectivity lexicon: ternary (strongsubj,weaksubj,not_member) {6}
PSC : the presence of the weak and strong subjectivity clues in the local context of 3 words: binary {4}

Figure 8: Subjectivity features for SWSD

training. Such an approach has the major disadvantage that we will need fully sense-tagged data to train our classifiers. In comparison, the former approach relies on subjectivity sense tagged data for training.

3.4 EXPERIMENTAL DESIGN

This section describes experiments on supervised SWSD based on expert annotations. Section 3.4.1 presents how we derive our training and test data for SWSD from SENSEVAL sense tagged corpora. In Section 3.4.2, we evaluate a SWSD module and compare its performance to conventional fine-grained WSD. Section 3.4.3 shows a simple rule-based approach to enable SWSD integration and evaluates the effect of SWSD on contextual subjectivity

analysis.

3.4.1 Data Creation

Our target words are members of the subjectivity lexicon introduced in section 2.1.1, because we know these words have subjective usages – since they appear in a subjectivity lexicon – and since the subjectivity analysis systems, to which we apply our SWSD module, rely on this lexicon.

The training and test data for SWSD consists of word instances in a corpus labeled as S or O, indicating whether they are used with a subjective or objective sense. Because we do not have data labeled with the S/O coarse-grained senses, we combine two types of sense annotations: (1) labels of senses within a dictionary as S or O (i.e., subjectivity sense labels), and (2) sense tags of word instances in a corpus (i.e., sense-tagged data). The subjectivity sense labels are used to collapse the sense labels in the sense-tagged data into the two new senses, S and O. This allows us to compare SWSD performance to WSD performance directly on the same dataset.

Our sense-tagged data are the lexical sample corpora (training and test data) from SENSEVAL I [Kilgarriff and Palmer, 2000], SENSEVAL II [Preiss and Yarowsky, 2001], and SENSEVAL III [Mihalcea and Edmonds, 2004]. We selected all of the SENSEVAL words that are also in the subjectivity lexicon, and labelled their dictionary senses as S, O, or B according to the annotation scheme described above in section 2.3. After excluding the senses labeled B (a total of 10 senses), we use remaining sense labels to collapse fine-grained senses in the SENSEVAL corpora. Among the words, we found that 11 are not ambiguous - either they have only S or only O senses (in the corresponding sense inventory), or the senses of their instances in the SENSEVAL data are all S or all O. So as not to inflate our results, we removed those 11 from the data, leaving 39 words: 9 words (64 senses) from the

SENSEVAL I, 18 words (201 senses) from the SENSEVAL II, and 12 words (107 senses) from the SENSEVAL III corpus. In total, we have subjectivity sense-tagged data for 39 subjectivity clues. From now on, we will refer to this expert subjectivity sense-tagged corpus as the senSWSD corpus. Overall, we can disambiguate 39 subjectivity clues from the lexicon which represents the coverage of the SWSD system trained on senSWSD.

3.4.2 In Vivo Evaluation

Our aim is to show that SWSD is a feasible task and that the subjectivity sense information provided by SWSD is much more reliable than the fine-grained sense information provided by WSD. For this purpose, we evaluate a SWSD system – based on coarse-grained training – on the senSWSD dataset and compare its performance to a WSD system on the same dataset via 10-fold cross validation experiments. Both SWSD and WSD systems utilize the WSD features introduced in Section 3.3.1.

In addition, we also evaluate fine-grained training approach for SWSD and compare it to coarse-grained training approach. Note that, although generally in the SENSEVAL datasets, training and test data are provided separately, a few target words from SENSEVAL I do not have both training and testing data. Thus, we opted to combine the training and test data into one dataset, and then perform 10-fold cross validation experiments.

As our classifier, we use the SVM classifier from the Weka package (Witten and Frank., 2005) with its default settings. We are interested in how well the system would perform on more and less ambiguous words. Thus, we split the words into three subsets according to their majority-class baselines – [50%,70%) , [70%,90%), and [90%,100%).

3.4.2.1 Results on Coarse-grained Training Table 1 contains the cumulative results over the whole senSWSD dataset, as well as results for the subsets *S1*, *S2*, and *S3*. *Base*

	Base	Acc	SP	SR	SF	OP	OR	OF	IB	EB%
All	79.9	88.3	89.3	89.1	89.2	87.1	87.4	87.2	8.4	41.8
S1	57.9	80.7	81.1	78.3	79.7	80.2	82.9	81.5	22.8	54.2
S2	81.1	87.3	86.5	85.2	85.8	87.9	89	88.4	6.2	32.8
S3	95	96.4	96.5	99	97.7	96.3	87.8	91.8	1.4	28.0

Table 1: Results of SWSD with coarse-grained training on senSWSD.

stands for the majority-class baseline. *Acc* is accuracy. *SP*, *SR*, and *SF* are subjective precision, recall and F-measure respectively – analogous for the objective class *O*. *IB* is improvement in accuracy over the baseline. *EB* is percent error reduction in accuracy.

In our evaluation, we see that the improvement for SWSD over the majority-class baseline is especially high for the less skewed set, S1. This is very encouraging because these words are the more ambiguous words, and thus are the ones that most need SWSD (assuming the SENSEVAL priors are similar to the priors in the target corpus). The average error reduction over baseline for S1 words is 54.2%. Even for the more skewed sets S2 and S3, reductions are 32.8% and 28.0%, respectively, with an overall reduction of 41.8%.

To compare SWSD with WSD, we re-run the 10-fold cross validation experiments, but this time using the original sense labels, rather than subjective sense and objective sense labels. The accuracy is 67.9%, much lower than the accuracy of SWSD (88.3%). The error reduction over the baseline for WSD is 18.9%, where SWSD provides an error reduction of 41.8%.

The positive results provide evidence that SWSD is a feasible variant of WSD, and that the S/O sense groupings are natural ones, since the system is able to learn to distinguish between them with high accuracy.

	Base	Acc	SP	SR	SF	OP	OR	OF	IB	EB%
All	79.9	86.3	87.1	88.3	87.7	85.4	84.0	84.7	6.4	31.8
S1	57.9	78.0	78.6	81.8	80.2	77.3	73.6	75.4	20.1	47.7
S2	81.1	85.0	83.5	83.1	83.3	86.3	86.6	86.5	4.9	25.9
S3	95	95.6	96.2	98.3	97.2	94.0	87.4	90.5	0.6	12

Table 2: Results of SWSD with fine-grained training on senSWSD.

3.4.2.2 Results on Fine-grained Training We repeat 10-fold cross validation experiment for SWSD with fine-grained training. Table 2 summarizes the results. We observe that coarse-grained training has better performance than fine-grained training. Overall, there is 2 percentage points difference in accuracy (Table 3). The difference is statistically significant at the $p < .05$ level according to a paired t-test.

3.4.2.3 Extending SWSD with Subjectivity Features In this section, we evaluate the effect of subjectivity features on the SWSD performance. For this purpose, we train two additional SWSD systems, one utilizing only subjectivity features introduced in Section 3.3.2 and the other one utilizing both WSD and subjectivity features. We evaluate both systems on the senSWSD dataset via 10-fold cross validation and compare the results to the original results.

Table 4 holds results for all three systems relying on different feature sets. We see that subjectivity features alone are not enough to have accurate SWSD. This result suggest that subjectivity of the surrounding context is not indicative of the sense subjectivity. Moreover, we do not see any improvement when we combine wsd features with subjectivity features.

	Base	Fine	Coarse
All	79.9	86.3	**88.3**
S1	57.9	78.0	**80.7**
S2	81.1	85.0	**87.3**
S3	95	95.6	**96.4**

Table 3: Results of SWSD: fine-grained training vs. coarse-grained training

3.4.3 In Vitro Evaluation

This section gives details on the conducted experiments to test the following hypothesis that SWSD can be exploited to improve the performance of contextual subjectivity analysis systems. We show in section 3.1, that there is a great deal of subjectivity sense ambiguity in a standard subjectivity-annotated corpus (MPQA) and then in section 3.4.2 that SWSD is a feasible task. We now turn to exploiting the results of SWSD to automatically recognize subjectivity in the MPQA Corpus. A motivation for using the MPQA Corpus is that many types of classifiers have been evaluated on it. We exploit SWSD in several contextual opinion analysis systems, comparing the performance of sense-aware and non-sense-aware versions. They are all variations of components of the OpinionFinder opinion recognition system.[1]

3.4.3.1 MPQA Coverage The SWSD system trained on the senSWSD dataset can disambiguate 39 target words, which have 723 instances in the MPQA Corpus. We refer to this subset of the MPQA Corpus as senMPQA. This subset makes up the coverage of the SWSD system evaluated in this section. Thus, we evaluate the effect of SWSD on contextual subjectivity analysis on senMPQA dataset. We integrate SWSD to two expression-level

[1]Available at http://www.cs.pitt.edu/opin

Feature Set	Acc
Base	79.9
WSD	**86.3**
WSD+Subj	85.7
Subj	80.8

Table 4: Results of SWSD: effect of subjectivity features on SWSD

contextual classifiers: (1) contextual polarity classifier labeling clue instances in text as contextually negative/positive/neutral, (2) contextual S/O classifier labeling clue instances in text as contextually subjective/objective. We incorporate SWSD information into these contextual subjectivity classifiers in a straight-forward fashion: outputs are modified according to simple, intuitive rules. In addition, we also integrate SWSD to a sentence-level rule-based classifier which labels sentences as subjective or objective with high precision and low recall. We compare original classifiers with sense-aware versions on senMPQA and draw conclusions according to McNemar's test for statistical significance [Dietterich, 1998].

Note that, for the SWSD experiments, the number of words does not limit the amount of data, as SENSEVAL provides data for each word. However, the only parts of the MPQA corpus for which SWSD could affect performance is the subset containing instances of the words in the SWSD system's coverage. Thus, for the classifiers in this section, the data used is the *SenMPQA* dataset, which consists of the sentences in the MPQA Corpus that contain at least one instance of the 39 keywords. There are 689 such sentences (containing, in total, 723 instances of the 39 keywords).

	Acc	OP	OR	OF	SP	SR	SF
O_{RB}	27	50	4.1	7.6	92.7	36	51.8
SE	28.3	62.1	9.3	16.1	92.7	35.8	51.6
RE	27.6	48.4	7.7	13.3	92.6	35.4	51.2

Table 5: Effect of SWSD on the Rule-based Classifiers.

3.4.3.2 Rule-based Classifier We first apply SWSD to a rule-based classifier from [Riloff and Wiebe, 2003]. The classifier, which is a sentence-level *S/O* classifier, has low subjective and objective recall but high subjective and objective precision. It is useful for creating training data for subsequent processing by applying it to large amounts of unannotated data.

The classifier is a good candidate for directly measuring the effects of SWSD on contextual subjectivity analysis, because it classifies sentences only by looking for the presence of subjectivity keywords. Performance will improve if false hits can be ignored.

The classifier labels a sentence as *S* if it contains two or more *strongsubj* clues. On the other hand, it considers three conditions to classify a sentence as *O*: there are no *strongsubj* clues in the current sentence, there are together at most one *strongsubj* clue in the previous and next sentence, and there are together at most 2 *weaksubj* clues in the current, previous, and next sentence.

The rule-based classifier is made sense aware by making it blind to the target word instances labeled *O* by the SWSD system, as these represent false hits of subjectivity keywords. We compare this sense-aware method (*SE*), with the original classifier (O_{RB}), in order to see if SWSD would improve performance. We also built another modified rule-based classifier *RE* to demonstrate the effect of randomly ignoring subjectivity keywords. *RE* ignores a

keyword instance randomly with a probability of 0.429, the expected value of false hits in the MPQA corpus. The results are listed in Table 5.

The rule-based classifier looks for the presence of the keywords to find subjective sentences and for the absence of the keywords to find objective sentences. It is obvious that a variant working on fewer keyword instances than O_{RB} will always have the same or higher objective recall and the same or lower subjective recall than O_{RB}. That is the case for both *SE* and *RE*. The real benefit we see is in objective precision, which is substantially higher for *SE* than O_{RB}. For our experiments, *OP* gives a better idea of the impact of SWSD, because most of the keyword instances SWSD disambiguates are *weaksubj* clues, and *weaksubj* keywords figure more prominently in objective classification. On the other hand, *RE* has both lower *OP* and *SP* than O_{RB}.

The overall low accuracy for all systems is due to the fact that all "unknown" predictions are considered false. Both *SE* and *RE* have higher accuracy than O_{RB}, because they both make less "unknown" predictions. The improvement in accuracy for *SE* is slightly better than *RE*. Although we think that the benefit with this classifier is reflected in objective precision, the larger improvement in accuracy for *SE* also indicates that *SE* is ignoring the right keyword instances - false hits. We cannot provide a significance test for accuracy, as we do for the other classifiers, because there are two gold labels (subj/obj) but three predicted labels (subj/obj/unknown).

These findings suggest that SWSD does a good job on disambiguating keyword instances in MPQA,[2] and demonstrates a positive impact of SWSD on sentence-level subjectivity classification.

[2]which we cannot evaluate directly, as MPQA is not sense tagged.

3.4.3.3 Contextual Subjective/Objective Classifier We now move to more fine-grained expression-level subjectivity classification. Since sentences often contain multiple subjective expressions, expression-level classification is more informative than sentence-level classification.

The contextual subjective/objective classifier is an implementation of the neutral/polar supervised classifier of (Wilson et al., 2005a) (using the same features), except that the classes are S/O rather than neutral/polar. These classifiers label instances of lexicon entries. The gold standard is defined on the MPQA Corpus as follows: If an instance is in a subjective expression, it is contextually subjective. If the instance is in an objective expression, it is contextually objective. We evaluate the system on the 723 clue instances in the SenMPQA dataset.

We incorporate SWSD information into the contextual subjectivity classifier in a straightforward fashion: the output is modified according to simple and intuitive rules in a post-processing step. Figure 9 demonstrates the general approach. Our strategy is defined by the relation between sense subjectivity and contextual subjectivity as explained in section 3.2 and involves two rules, *R1* and *R2*. We know that a keyword instance used with a S sense must be in a subjective expression. *R1* is to simply trust SWSD: If the contextual classifier labels an instance as O, but SWSD determines that it has an S sense, then *R1* flips the contextual classifier's label to S.

Things are not as simple in the case of O senses, since they may appear in both subjective and objective expressions. We will state *R2*, and then explain it: If the contextual classifier labels an instance as S, but (1) SWSD determines that it has an O sense, (2) the contextual classifier's confidence is low, and (3) there is no other subjective keyword in the same expression, then *R2* flips the contextual classifier's label to O. First, consider confidence: though a keyword with an O sense may appear in either subjective or objective expressions, it is more

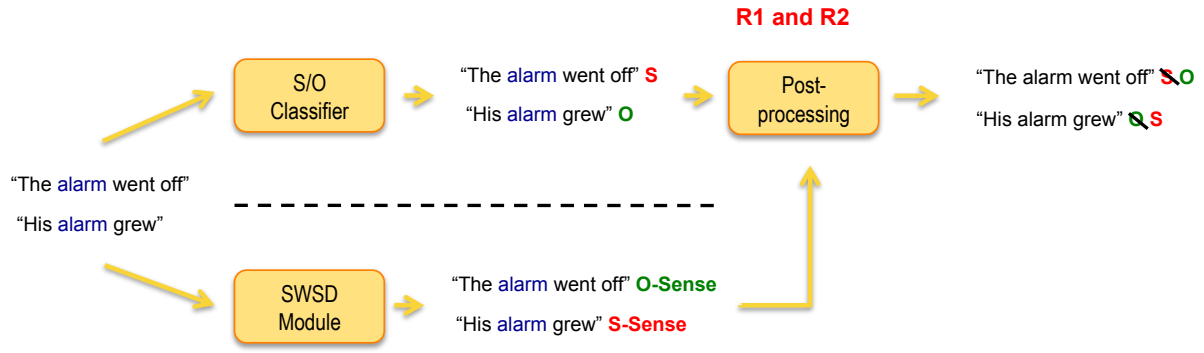

Figure 9: SWSD integration to contextual subjectivity classifier.

likely to appear in an objective expression. We assume that this is reflected to some extent in the contextual classifier's confidence. Second, if a keyword with an O sense appears in a subjective expression, then the subjectivity is not due to that keyword but rather due to something else. Thus, the presence of another lexicon entry "explains away" the presence of the O sense in the subjective expression, and we do not want SWSD to overrule the contextual classifier. Only when the contextual classifier is not certain and only when there is not another keyword does *R2* flip the label to O.

Our definition of low confidence is in terms of the label weights assigned by BoosTexter [Schapire and Singer, 2000], which is the underlying machine learning algorithm of the contextual classifiers. We use the difference between the largest weight of any label and the second largest label weight as a measure of confidence, as suggested in the BoosTexter documentation. For the experiments on the *subjective/objective classifier*, we adopt the threshold determined for the *neutral/polar classifier* in Section 3.4.3.4. Note, that we do not experiment with other conditions than those incorporated in the rules.

	Acc	OP	OR	OF	SP	SR	SF
$O_{S/O}$	75.4	68	62.9	65.4	79.2	82.7	80.9
R1	77.7	75.5	58.8	66.1	78.6	88.8	83.4
R2	79	67.3	83.9	74.7	89	76.1	82
CM	81.3	72.5	79.8	75.9	87.4	82.2	84.8

Table 6: Effect of SWSD on the Subjective/Objective Classifier.

We compare the performance of the original system ($O_{S/O}$) and three sense-aware variants: one using only *R1*, one using only *R2*, and one using both (*CM*). The results are summarized in Table 6. The original classifier has an accuracy of 75.4%. The *R1* variant shows an improvement of 2.3 percentage points in accuracy to 77.7 (a 9.4% error reduction). The R2 variant shows an improvement of 3.6 percentage points in accuracy to 79 (a 14.6% error reduction). Applying both rules (*CM*) gives an improvement of 5.9 percentage points in accuracy to 81.3 (a 24% error reduction).

In our case, a paired t-test is not appropriate to measure statistical significance, as we are not doing multiple runs. Thus, we apply McNemar's test, which is a non-parametric method for algorithms that can be executed only once, meaning training once and testing once [Dietterich, 1998]. For *R1*, the improvement in accuracy is statistically significant at the $p < .05$ level. For *R2* and *CM*, the improvement in accuracy is statistically significant at the $p < .01$ level. Moreover, in all cases, we see improvement in both objective and subjective F-measure.

3.4.3.4 Contextual Polarity Classifier In this section, we apply SWSD to contextual polarity classification (positive /negative/ neutral), in the hope that avoiding false hits

of subjectivity keywords will also lead to performance improvement in contextual polarity classification.

We use an implementation of the classifier of (Wilson et al., 2005a). This classifier labels instances of lexicon entries. The gold standard is defined on the MPQA Corpus as follows: If an instance is in a positive subjective expression, it is contextually positive *(Ps)*; if in a negative subjective expression, it is contextually negative *(Ng)*; and if it is in an objective expression or a neutral subjective expression, then it is contextually N(eutral). As above, we evaluate the system on the keyword instances in the SenMPQA dataset.

Wilson et al. use a two step approach. The first step classifies keyword instances as being in a polar (positive or negative) or a neutral context. The first step is performed by the neutral/polar classifier mentioned above in 3.4.3.3. The second step decides the contextual polarity (positive or negative) of the instances classified as polar in the first step, and is performed by a separate classifier.

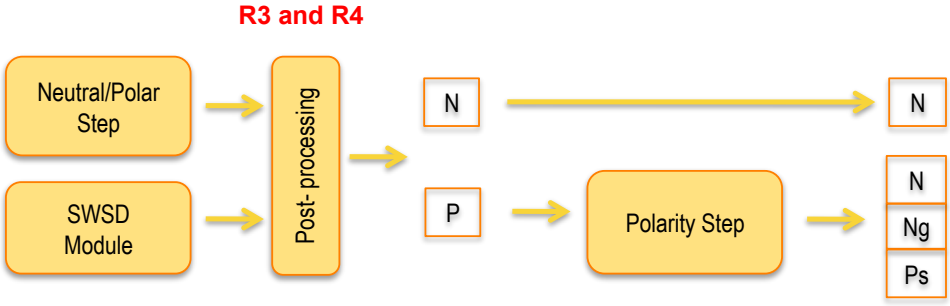

Figure 10: SWSD integration to contextual polarity classifier.

To make a sense-aware version of the system, again we use rules to modify of the output of the neutral/polar classifier. *R3* flips neutral/polar classifier's output from *N* to *P*. It is almost analogous to *R1*. There is a small difference. We cannot simply trust SWSD and flip the output to *P* when it labels a keyword as having an S sense, because an S sense might be in a *N(eutral)* expression (since there are neutral subjective expressions). But, an S sense

	Acc	NP	NR	NF	PP	PR	PF
$O_{N/P}$	79	81.5	92.5	86.7	65.8	40.7	50.3
R3	70	83.7	73.8	78.4	44.4	59.3	50.8
R4	81.6	81.7	96.8	88.6	81.1	38.6	52.3

Table 7: Effect of SWSD on the Neutral/Polar Classifier.

is more likely to appear in a *P(olar)* expression. Thus, we consider confidence (rule *R3*): If the contextual classifier labels an instance as *N*, but SWSD determines it has an S sense and the contextual classifier's confidence is low, then *R3* flips the contextual classifier's label to *P*.

Rule *R4* is analogous to *R2* in the previous section: If the contextual classifier labels an instance as P, but (1) SWSD determines that it has an O sense, (2) the contextual classifier's confidence is low, and (3) there is no other subjective keyword in the same expression, then *R4* flips the contextual classifier's label to N.

As mentioned before, our definition of "low confidence" depends on the difference between the largest weight of any label and the second largest label weight assigned by BoosTexter [Schapire and Singer, 2000]. We tried three thresholds: 0.0007, 0.0008, and 0.0009. The difference in accuracy when using different thresholds is slight: 0.0007 and 0.0009 both give 81.5 accuracy compared to 81.6 accuracy for 0.0008. Here, we report results for 0.0008. Note that exactly the same significance test results apply to all the thresholds tried and we do not try other conditions for both rules.

We compare the performance of the original neutral/polar classifier ($O_{N/P}$) and sense-aware variants using *R3* and *R4*. The results are summarized in Table 7. *PP*, *PR*, and

	Acc	NP	NR	NF	NgP	NgR	NgF	PsP	PsR	PsF
$O_{Ps/Ng/N}$	77.6	80.9	94.6	87.2	60.4	29.4	39.5	52.2	32.4	40
R4	80.6	81.2	98.7	89.1	82.1	29.4	43.2	68.6	32.4	44

Table 8: Effect of SWSD on the Contextual Polarity Classifier.

PF stand for polar precision, recall and F-measure – analogous for the neutral class *N*. We see that only *R4* improves performance. This is consistent with the finding in (Wilson et al., 2005a) that most errors are caused by subjectivity keywords with non-neutral prior polarity appearing in phrases with neutral contextual polarity. *R4* targets these cases. It is promising to see that SWSD provides enough information to fix some of them. There is a 2.6 percentage point improvement in accuracy from 70% to 81,6% (a 12.4% error reduction). The improvement in accuracy is statistically significant at the $p < .01$ level with McNemar's test. The improvement in accuracy is accompanied by improvements in both neutral and polar F-measure.

We want to see if the improvements in the first step of Wilson et al's system will propagate to the output of the second step, yielding an overall improvement in positive /negative/neutral (*Ps*/*Ng*/*N*) classification. The sense-aware variant of the overall two-part system is the same as the original except that we apply *R4* to the output of the first step (flipping some of the neutral/polar classifier's *P* labels to *N*). Thus, since the second step in Wilson et al.'s classifier processes only those instances labelled *P* in the first step, in the sense-aware system, fewer instances are passed from the first to the second step. Table 8 holds results. *NP*, *NR*, and *NF* stand for neutral precision, recall and F-measure – analogous for the negative (*Ng*) and positive (*Ps*) class Note that these results are for the entire SenMPQA dataset, not just those labeled *P* in the first step. We see that the accuracy improves 3 percentage

points from 77.6 to 80.6 (a 13.4% error reduction). The improvement in accuracy is statistically significant at the $p < .01$ level with McNemar's test. We see the real benefit when we look at the precision of the positive and negative classes. Negative precision goes from 60.4 to 82.1 and positive precision goes from 52.2 to 68.6, with no loss in recall. This is evidence that the SWSD system is doing a good job of removing some false hits of subjectivity clues that harm the original version of the system.

3.5 SUMMARY AND DISCUSSION

In this chapter, we introduced the task of subjectivity word sense disambiguation (SWSD), and evaluated a supervised method inspired by research in WSD. The system achieves high accuracy, especially on highly ambiguous words. The results provide evidence for our first hypothesis:

> **Hypothesis 1**: S/O sense groupings are natural and both groups can be disambiguated accurately by a supervised model.

We compared the SWSD accuracy to the WSD accuracy on the same dataset. SWSD is substantially better than WSD. Moreover, SWSD provides in total two sense types S/O. Thus, integration of the sense information to the underlying opinion system is uniform. It does not suffer from the sparsity problem that will form by utilizing fine-grained senses, which grow linearly in the number of the subjectivity clues we want to disambiguate. We investigate two methods to train our SWSD classifiers: (1) the coarse-grained training and (2) the fine-grained training. We observe that the coarse-grained training results in 15% error reduction over the fine-grained training. This experiment also allows a more systematic comparison of the sense information provided by SWSD to the sense information provided by WSD. Even

after collapsing predicted fine-grained senses to coarse-grained senses, information provided by a full WSD system is not as reliable as the system trained on coarse-grained senses. Moreover, the fine-grained training has the disadvantage that it requires fully sense-tagged data. The annotation effort to generate coarse-grained annotations is smaller. The results support our second hypothesis:

Hypothesis 2: The subjectivity sense information provided by SWSD is more reliable than the fine-grained sense information provided by WSD.

We explored the promise of SWSD for contextual subjectivity analysis. We showed that a subjectivity lexicon can have substantial coverage of the subjective expressions in the corpus, yet still be responsible for significant sense ambiguity. This demonstrates the potential benefit to opinion analysis of performing SWSD. We then exploited SWSD in several contextual subjectivity analysis systems, including positive/negative/neutral sentiment classification. Improvements in performance were realized for all of the systems. These results are evidence for our third hypothesis:

Hypothesis 3: SWSD can be exploited to improve the performance of contextual subjectivity analysis systems via sense-aware analysis.

In addition, we evaluate the effect of subjectivity features on the SWSD performance. The results show that subjectivity features alone are not enough to have accurate SWSD suggesting that subjectivity of the surrounding context is not indicative of the sense subjectivity. Moreover, we do not see any improvement when we combine wsd features with subjectivity features.

In this chapter, we exclusively relied on expert annotations, which are limited in availability. Thus, the coverage of our SWSD system was not high. In addition, we utilized

manually crafted rules to integrate SWSD into the contextual subjectivity analysis, which are ad hoc. We address these shortcomings in later chapters.

3.6 RELATED WORK

Several researchers exploit lexicons of subjectivity bearing words for contextual subjectivity analysis. These systems typically look for the presence of lexicon clues in the text to be analysed. There are two general approaches to utilize these lexicons. They are either used in a knowledge-based approach where the information about all clue instances are aggregated to the enclosing text (e.g. [Turney, 2002b, Yu and Hatzivassiloglou, 2003, Kim and Hovy, 2004, Hu and Liu, 2004, Ding et al., 2008, Zhai et al., 2011]) or they are utilized in a supervised setting where they become features for a machine learning algorithm (e.g. [Riloff and Wiebe, 2003, Whitelaw et al., 2005, Wilson et al., 2005, Agarwal et al., 2009, Jiang et al., 2011]). Both types of systems that rely on subjectivity lexicons can benefit from SWSD via sense-aware analysis.

One related line of research is to automatically assign subjectivity and/or polarity labels to word senses in a dictionary. [Esuli and Sebastiani, 2006b] and [Andreevskaia and Bergler, 2006] are the most prominent works on polarity labelling of word senses. Both works start with positive and negative seed sets and expand polarity by traversing specific links in WordNet. An extensive amount of work is also done on assigning subjectivity labels to word senses. [Wiebe and Mihalcea, 2006] use a corpus based approach where subjectivity labels are assigned based on a set of distributionally similar words in the MPQA Corpus. [Gyamfi et al., 2009] and [Su and Markert, 2008] use supervised classifiers relying on features defined on WordNet for subjectivity labelling. All these methods have the aim to assign labels to word senses in a sense inventory automatically. In contrast, we automatically assign labels to

word instances in context. A common point we have with [Wiebe and Mihalcea, 2006,Gyamfi et al., 2009] is the annotation schema we use to manually label senses of a word, which is also similar to the schema utilized by [Su and Markert, 2008]. Moreover, our work complements findings in [Wiebe and Mihalcea, 2006] and [Gyamfi et al., 2009]. [Wiebe and Mihalcea, 2006] demonstrate that subjectivity is a property that can be associated with word senses. We show that subjectivity provides a natural grouping of word senses. [Wiebe and Mihalcea, 2006] also demonstrate that subjectivity can be utilized to improve WSD. We show that a coarse-grained WSD variant (SWSD) improves contextual subjectivity analysis. [Gyamfi et al., 2009] shows in a study that even in subjectivity lexicons, a large proportion – almost 50% – of the senses are objective. We demonstrate that ambiguity is also prevalent in a corpus.

Recently, some researchers have exploited fine grained WSD in methods for subjectivity analysis. They are [Ar et al., 2011], [Martín-Wanton et al., 2010], and [Rentoumi et al., 2009]. Their approaches are very different from ours. [Ar et al., 2011] benefit from WSD for document-level polarity classification. They represent a document as a bag-of-senses instead of a bag-of-words. This means a document consisting of words gets mapped to a document consisting of corresponding word senses. In contrast to our work, [Ar et al., 2011] do not disambiguate the subjectivity or polarity of a word explicitly. To be specific, their approach lacks any special treatment of the underlying task that is subjectivity analysis. When the bag-of-senses representations are created by oracle information, the system shows a large improvement over the unigram bag-of-words model. But, when the bag-of-senses representation is created automatically by fine-grained WSD, the improvement is not significant. The results indicate that fine-grained WSD is not accurate enough for this kind of application, even though [Ar et al., 2011] concentrate on a single domain – travel reviews – and utilize domain-specific WSD. In contrast, our work achieves significant improvement

with automatic coarse-grained WSD. Our work is also more general in the sense that we do not restrict ourselves to a single domain. Another difference is that our target tasks are expression-level. [Rentoumi et al., 2009] deals with polarity classification of news headlines. [Rentoumi et al., 2009] first determines the sense of a word instance and then assign polarity to the sense according to a polarity lexicon. They train a supervised classifier on sense-level polarity information. Their approach is limited only to figurative expressions where our approach is more general and can be applied to arbitrary text. [Martín-Wanton et al., 2010] deals with polarity classification of short newspaper quotes. They follow a similar approach to [Rentoumi et al., 2009]. The only difference is that they aggregate the polarity information of single word instances (senses) to the enclosing quote in an unsupervised way. Both [Martín-Wanton et al., 2010], and [Rentoumi et al., 2009] disambiguate words for their polarity explicitly. Our work differs in that we disambiguate lexicon clues for their subjectivity. None of this previous work investigates performing a coarse-grained variation of WSD such as SWSD to improve their application results, as we do in our work.

A notable exception is [Su and Markert, 2010], who exploit SWSD to improve the performance on a contextual NLP task, as we do. While our task is subjectivity analysis, their task is English-Chinese lexical substitution. [Su and Markert, 2010] adopt our definition of SWSD. As we do, they manually annotate word senses, and exploit SENSEVAL data as training data for SWSD. They do not directly annotate words in context with S and O labels, as we do in our work. Further, they do not separately evaluate a SWSD system component. They incorporate SWSD information as a single feature to the base lexical substitution classifier, as we do in one of our integration methods.

Many WSD systems use WordNet as their sense inventory. Although WordNet is an established lexical resource, the fine grained sense distinctions in WordNet create an upperbound for the achievable performance of WSD systems. [Palmer et al., 2004] reports

an inter-annotator agreement of 72.5% for the English all-words test set at SENSEVAL 3. Many groups worked on the grouping of WordNet senses. They aim for a more coarse-grained sense inventory to overcome performance shortcomings related to fine-grained sense distinctions. [Mihalcea and Moldovan, 2001] derived semantic and probabilistic rules to group similar senses in WordNet in an unsupervised approach. Their efforts resulted in a new version of WordNet with 26% less polysemy and minimal error rate as measured on a sense tagged corpus. [Navigli, 2006] and [Palmer et al., 2004] map WordNet senses to a coarse-grained sense inventory reporting improved inter-annotator agreement and system performance on the coarse grained level. In the former work, the mapping is done via an automatic system utilizing Lesk-like and complex semantic features. In the latter work, the mapping is done manually - [Snow et al., 2007] integrates many previously proposed features based on WordNet similarity metrics, corpus statistics, and mappings to existing lexical resources, building a supervised system for merging WordNet senses. The OntoNotes project [Pradhan and Xue, 2009] is another important work in this field . Linguists and annotators work together to group WordNet senses with the goal to have high-interannoter agreement. Our work is similar in the sense that we reduce all senses of a word to two senses (S/O). The difference is the criterion driving the grouping. Related work concentrates on syntactic and semantic similarity between senses to group them. They usually move away from polysemous sense distinctions and focus more on homonymous sense distinctions, which are easy to make. In contrast, our grouping is driven by subjectivity, with a specific application area in mind, namely subjectivity analysis.

4.0 NON-EXPERT ANNOTATIONS

In chapter 3, we see that supervised SWSD achieves high accuracy especially on highly ambiguous words, and substantially outperforms WSD on the same dataset. More importantly, the integration of SWSD results in substantial improvement for contextual subjectivity analysis. Although the results are very promising, there are three shortcomings. First, we are not able to apply SWSD to contextual opinion analysis on a large scale, due to a shortage of annotated data. Two questions arise: is it feasible to obtain greater amounts of the needed data, and do SWSD performance improvements on contextual opinion analysis hold on a larger scale. Second, the annotations in chapter 3 are piggy-backed on SENSEVAL sense-tagged data, which are fine-grained word sense annotations created by trained annotators. A concern is that SWSD performance improvements on contextual opinion analysis can only be achieved using such fine-grained expert annotations, the availability of which is limited. Third, in chapter 3, we define manual rules to integrate SWSD into contextual subjectivity analysis. Although these rules have the advantage that they transparently show the effects of SWSD, they are somewhat ad hoc. Likely, they are not optimal and are holding back the potential of SWSD to improve contextual opinion analysis.

In this chapter, we investigate (1) the feasibility of obtaining a substantial amount of annotated data and expand the coverage of our SWSD system (2) whether performance improvements on contextual opinion analysis can be realized on a larger scale, and (3) whether those improvements can be realized with subjectivity sense tagged data that is not

built on expert full-inventory sense annotations. For this purpose, we obtain non-expert annotations via *Amazon Mechanical Turk* (MTurk), a cheap and fast alternative to expert annotations.

In section 4.1, we give general background information on Amazon Mechanical Turk (MTurk). Section 4.2 describes how we set up the subjectivity sense tagging for the MTurk environment and how we evaluate the quality of non-expert annotations. In Section 4.3, we build a SWSD system on non-expert annotations and exploit non-expert SWSD for sense-aware subjectivity analysis. In the same section, we also introduce new integration methods. The research presented in this chapter is published in [Akkaya et al., 2010, Akkaya et al., 2011]

4.1 AMAZON MECHANICAL TURK

Amazon Mechanical Turk (MTurk)[1] is a marketplace for so-called "*human intelligence tasks*" or HITs. MTurk has two kinds of users: *providers* and *workers*. Providers create HITs using the Mechanical Turk API and, for a small fee, upload them to the HIT database. Workers search through the HIT database, choosing which to complete in exchange for monetary compensation. Anyone can sign up as a provider and/or worker. Each HIT has an associated monetary value, and after reviewing a worker's submission, a provider may choose whether to accept the submission and pay the worker the promised sum or to reject it and pay the worker nothing. HITs typically consist of tasks that are easy for humans but difficult or impossible for computers to complete quickly or effectively, such as annotating images, transcribing speech audio, or writing a summary of a video.

Recently researchers have been investigating Amazon Mechanical Turk (MTurk) as a

[1] http://mturk.amazon.com

source of non-expert natural language annotation [Kaisser and Lowe, 2008, Mrozinski et al., 2008]. The annotations obtained from MTurk workers are noisy by nature, because MTurk workers are not trained for the underlying annotation task. It is understandable that not every worker will provide high-quality annotations, depending on their background and interest. Unfortunately, some MTurk workers do not follow the annotation guidelines and carelessly submit annotations in order to gain economic benefits with only minimal effort. We define this group of workers as spammers. One challenge for requesters using MTurk is that of filtering out spammers and other workers who consistently produce low-quality annotations. It is essential to distinguish between workers as well-meaning annotators and workers as spammers who should be filtered out as a first step when utilizing MTurk.

MTurk provides several types of built-in statistics, known as qualifications, in order to allow requesters to restrict the range of workers who can complete their tasks. One such qualification is approval rating, a statistic that records a worker's ratio of accepted HITs compared to the total number of HITs submitted by that worker. Providers can require that a worker's approval rating be above a certain threshold before allowing that worker to submit one of his/her HITs. Country of residence and lifetime approved number of HITs completed also serve as built-in qualifications that providers may check before allowing workers to access their HITs.[2] In addition, MTurk allows providers to define their own qualifications. Typically, provider-defined qualifications are used to ensure that HITs which require particular skills are only completed by qualified workers. In most cases, workers acquire provider-defined qualifications by completing an online test. Mturk also provides a mechanism by which multiple unique workers can complete the same HIT. The number of times a HIT is to be completed is known as the number of assignments for the HIT. By

[2]According to the terms of use, workers are prohibited from having more than one account, but to our knowledge there is no method in place to enforce this restriction. Thus, a worker with a poor approval rating could simply create a new account, since all accounts start with an approval rating of 100%.

having multiple workers complete the same HIT, techniques such as majority voting among the submissions can be used to aggregate the results for some types of HITs, resulting in a higher-quality final answer. Previous work [Snow et al., 2008] demonstrates that aggregating worker submissions often leads to an increase in quality.

4.2 AMAZON MECHANICAL TURK FOR SWSD

4.2.1 Subjectivity Sense Tagging via Amazon Mechanical Turk

In this section, we describe how subjectivity word sense tagging is done by MTurk workers. We try to keep the annotation task for the worker as simple as possible. Thus, we do not directly ask them if the instance of a target word has a subjective or an objective sense (without any sense inventory), because the concept of subjectivity is fairly difficult to explain to someone who does not have any linguistics background. Instead we show MTurk workers two sets of senses – one subjective set and one objective set – for a specific target word and a text passage in which the target word appears. Their job is to select the set that best reflects the meaning of the target word in the text passage. The specific sense set automatically gives us the subjectivity label of the instance. This makes the annotation task easier for them as [Snow et al., 2008] shows that WSD can be done reliably by MTurk workers. This approach presupposes a set of word senses that have been annotated as subjective or objective. The annotation of senses in a dictionary for subjectivity is not difficult for an expert annotator. Moreover, it needs to be done only once per target word, allowing us to collect hundreds of subjectivity labelled instances for each target word through MTurk.

In this annotation task, we do not inform the MTurk workers about the nature of the sets. This means the MTurk workers have no idea that they are annotating subjectivity

Sense_Set1 (Subjective)

{ look, **appear**, seem } – give a certain impression or have a certain outward aspect; "She seems to be sleeping"; "This appears to be a very difficult problem"; "This project looks fishy"; "They appeared like people who had not eaten or slept for a long time"
{ **appear**, seem } – seem to be true, probable, or apparent; "It seems that he is very gifted"; "It appears that the weather in California is very bad"

Sense_Set2 (Objective)

{ **appear** } – come into sight or view; "He suddenly appeared at the wedding"; "A new star appeared on the horizon"
{ **appear**, come_out } – be issued or published, as of news in a paper, a book, or a movie; "Did your latest book appear yet?"; "The new Woody Allen film hasn't come out yet"
{ **appear**, come_along } – come into being or existence, or appear on the scene; "Then the computer came along and changed our lives"; "Homo sapiens appeared millions of years ago"
{ **appear** } – appear as a character on stage or appear in a play, etc.; "Gielgud appears briefly in this movie"; "She appeared in 'Hamlet' on the London
{ **appear** } – present oneself formally, as before a (judicial) authority; "He had to appear in court last month"; "She appeared on several charges of theft"

Figure 11: Sense sets for target word "appear".

of senses; they are just selecting the set which contains a sense matching the usage in the sentence or being as similar to it as possible. This ensures that MTurk workers are not biased by the contextual subjectivity of the sentence while tagging the target word instance.

Below, we describe a sample annotation problem. An MTurk worker has access to two sense sets of the target word "appear" as seen in Figure 11. The information that the first sense set is subjective and second sense set is objective is not available to the worker.

The worker is presented with the following text passage holding the target word "appear". The worker should be able to understand that "appeared" refers to the outward impression given by "Charles". This use of appear is most similar to the first entry in sense set one; thus, the correct answer for this problem is Sense_Set-1.

(4.1) It's got so bad that I don't even know what to say. Charles **appeared** somewhat

embarrassed by his own behavior. The hidden speech was coming, I could tell.

4.2.2 Annotation Quality

This section gives details on the conducted experiments to test if built-in methods for annotation quality are enough to avoid spammers and if we can utilize MTurk to collect high-quality annotations for SWSD.

4.2.2.1 Experimental Design We chose randomly 8 target words that have a distribution of subjective and objective instances in senSWSD with less skew than 75%. That is, no more than 75% of a word's senses are subjective or objective. Our concern is that using skewed data might bias the workers to choose from the more frequent label without thinking much about the problem. Another important fact is that these words with low skew are more ambiguous and responsible for more false hits. Thus, these target words are the ones for which we really need subjectivity word sense disambiguation. For each of these 8 target words, we select 40 passages from senSWSD in which the target word appears, to include in our experiments. Table 9 summarizes the selected target words and their label distribution. In this table, frequent label percentage (FLP) represents the skew for each word. A word's FLP is equal to the percent of the senses that are of the most frequently occurring type of sense (subjective or objective) for that word.

We believe this annotation task is a good candidate for attracting spammers. This task requires only binary annotations, where the worker just chooses from one of the two given sets, which is not a difficult task. Since it is easy to provide labels, we believe that there will be a distinct line, with respect to quality of annotations, between spammers and mediocre annotators.

For our experiments, we created three different HIT groups each having different quali-

Word	FLP	Word	FLP
appear	55%	fine	72.5%
judgment	65%	solid	55%
strike	62.5%	difference	67.5%
restraint	70%	miss	50%
Average	62.2%		

Table 9: Frequent label percentages of the target words in the MTurk experiment.

fication requirements but sharing the same data. To be concrete, each HIT group consists of the same 320 instances: 40 instances for each target word listed in Table 9. Each HIT presents an MTurk worker with four instances of the same word in a text passage – this makes 80 HITs for each HIT group – and asks him to choose the set to which the activated sense belongs. We know for each HIT the mapping between sense set numbers and subjectivity. Thus, we can evaluate each HIT response on our gold-standard data senSWSD. We pay seven cents per HIT. We consider this to be generous compensation for such a simple task.

There are many builtin qualifications in MTurk. We concentrated only on three of them: location, HIT approval rate, and approved HITs, as discussed in Section 4.1. In our experience, these qualifications are widely used for quality assurance. As mentioned before, we created three different HIT groups in order to see how well different built-in qualification combinations do with respect to filtering spammers. These groups – starting from the least constrained to the most constrained – are listed in Table 10.

Group1 required only that the MTurk workers are located in the US. This group is the

Group1	Location: USA
Group2	Location: USA HIT Approval Rate > 96%
Group3	Location: USA HIT Approval Rate > 96% Approved HITs > 500

Table 10: Constraints for each HIT group.

least constrained one. *Group2* additionally required an approval rate greater than 96%. *Group3* is the most constrained one, requiring a lifetime approved HIT number to be greater than 500, in addition to the qualifications in Group1 and Group2.

We believe that neither location nor approval rate and location together is enough to avoid spammers. While being a US resident does to some extent guarantee English proficiency, it does not guarantee well-thought answers. Since there is no mechanism in place preventing users from creating new MTurk worker accounts at will and since all worker accounts are initialized with a 100% approval rate, we do not think that approval rate is sufficient to avoid serial spammers and other poor annotators. We hypothesize that the workers with high approval rate and a large number of approved HITs have a reputation to maintain, and thus will probably be careful in their answers. We think it is unlikely that spammers will have both a high approval rate and a large number of completed HITs. Thus, we anticipated that Group3's annotations will be of higher quality than those of the other groups.

Note that an MTurk worker who has access to the HITs in one of the HIT groups also has

access to HITs in less constrained groups. For example, an MTurk worker who has access to HITs in Group3 also has access to HITs in Group2 and Group1. We did not prevent MTurk workers from working in multiple HIT groups because we did not want to influence worker behavior, but instead simulate the most realistic annotation scenario.

In addition to the qualifications described above, we also required each worker to take a qualification test in order to prove their competence in the annotation task. The qualification test consists of 10 simple annotation questions identical in form to those present in the HITs. These questions are split evenly between two target words, "appear" and "restraint". There are a total of five subjective and five objective usages in the test. We required an accuracy of 90% in the qualification test, corresponding to a Kappa score of .80, before a worker was allowed to submit any of our HITs. If a worker failed to achieve a score of 90% on an attempt, that worker could try the test again after a delay of 4 hours.

We collected three sets of assignments within each HIT group. In other words, each HIT was completed three times by three different workers in each group. This gives us a total of 960 assignments in each HIT group. A total of 26 unique workers participated in the experiment: 17 in Group1, 17 in Group2 and 8 in Group3. As mentioned before, a worker is able to participate in all the groups for which he is qualified. Thus the unique worker numbers in each group does not sum up to the total number of workers in the experiment, since some workers participated in the HITs for more than one group. Figure 12 summarizes how workers are distributed between groups.

We are interested in how accurate the MTurk annotations are with respect to gold-standard data. We are also interested in how the accuracy of each group differs from the others. We evaluate each group itself separately on the gold-standard data. Additionally, we evaluate each worker's performance on the gold-standard data and inspect their distribution in various groups.

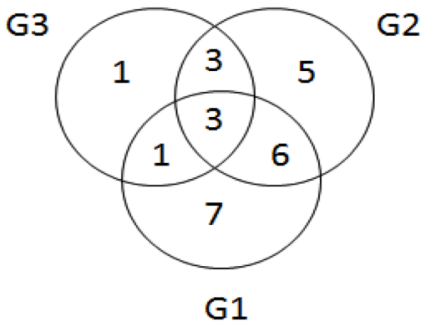

Figure 12: Venn diagram illustrating worker distribution.

4.2.2.2 Group Evaluation As mentioned in the previous section, we collect three annotations for each HIT. They are assigned to respective trials in the order submitted by the workers. The results are summarized in Table 11. Trials are labeled as T_X and MV is the majority vote annotation among the three trials. The final column contains the baseline agreement where a worker labels each instance of a word with the most frequent label of that word in the gold-standard data. High percentage agreement – frequent label percentage is 62.2 – and kappa scores in all groups and trials provide evidence that subjectivity word sense tagging can be done reliably by MTurk workers. This is very promising considering the low cost and short time required to obtain MTurk annotations.

When we compare groups with each other, we see that the best trial result is achieved in Group3. However, according to McNemar's test [Dietterich, 1998], there is no statistically significant difference between any trial of any group. On the other hand, the best majority vote annotation is achieved in Group2, but again there is no statistically significant difference between any majority vote annotation of any group. These results are surprising to us, since we do not see any significant difference in the quality of the data throughout different groups.

	Group3				Group2				Group1			
	T_1	T_2	T_3	MV	T_1	T_2	T_3	MV	T_1	T_2	T_3	MV
Accuracy	89.7	86.9	**86.6**	88.4	87.2	86.3	88.1	**90.3**	84.4	87.5	87.5	88.4
Kappa	.79	.74	.73	.77	.74	.73	.76	.81	.69	.75	.75	.77

Table 11: Accuracy and kappa scores for each group of workers.

4.2.2.3 Worker Evaluation In this section, we evaluate all 26 workers and group them as either spammers or well-meaning workers. All workers who deviate from the gold-standard by a large margin beyond a certain threshold will be considered to be spammers. As discussed in Section 4.2.2.1, we require all participating workers to pass a qualification test before answering HITs. Thus, we know that they are competent to do subjectivity sense annotations, and providing consistently erroneous annotations means that they are probably spammers. We think a kappa score of 0.6 is a good threshold to distinguish spammers from well-meaning workers. For this threshold, we had 2 spammers participating in Group1, 2 spammers in Group2 and 0 spammers in Group3. Table 12 presents spammer count and spammer percentage in each group for various threshold values. We see that Group3 has consistently fewer spammers and a smaller spammer percentage. The lowest kappa scores for Group1, Group2, and Group3 are .35, .40, and .69, respectively. The mean kappa scores for Group1, Group2, and Group3 are .73, .75, and .77, respectively.

These results indicate that Group3 is less prone to spammers, apparently contradicting Section 4.2.2.2. We see the reason when we inspect the data more closely. It turns out that spammers contributed in Group1 and Group2 only minimally. On the other hand there are two mediocre workers (Kappa of 0.69) who submit around 1/3 of the HITs in Group3. This

Threshold		0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75
Spammer Count	G1	2	2	2	2	2	4	7	9
	G2	1	2	2	2	2	3	5	8
	G3	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**2**
Spammer Percentage	G1	12%	12%	12%	12%	12%	24%	41%	53%
	G2	6%	12%	12%	12%	12%	12%	29%	42%
	G3	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**	**25%**	**25%**

Table 12: Spammer representation in groups.

behavior might be a coincidence. In the face of contradicting results, we think that we need a more extensive study to derive conclusions about the relation between spammer distribution and built-in qualification.

4.2.2.4 Learning Effect Another important question about MTurk workers is whether they learn to provide better annotations over time in the absence of any interaction and feedback. The presence of a learning effect may support working with the same workers over a long time and creating private groups of workers.

Expert annotators can learn to provide more accurate annotations over time. [Passonneau et al., 2006] reports a learning effect early in the annotation process. This might be due to the formal and informal interaction between annotators. Another possibility is that the annotators might get used to the annotation task over time. This is to be expected if there is not an extensive training process before the annotation takes place.

On the other hand, the MTurk workers have no interaction among themselves. They

do not receive any formal training and do not have access to true annotations except a few examples if provided by the requester. These properties make MTurk workers a unique annotation workforce. We are interested if the learning effect common to expert annotators holds in this unique workforce in the absence of any interaction and feedback. That may justify working with the same set of workers over a long time by creating private groups of workers.

We sort annotations of a worker after the submission date. This way, we get for each worker an ordered list of annotations. We split the list into bins of size 40 and we test for an increasing trend in the proportion of successes over time. We use the Chi-squared Test for binomial proportions [Rosner, 2006]. Using this test, we find that all of the p-values are substantially larger than 0.05. Thus, there is no increasing trend in the proportion of successes and no learning effect. This is true for both mediocre workers and very reliable workers. We think that the results may differ for harder annotation tasks where the input is more complex and requires some adjustment.

4.3 SWSD ON NON-EXPERT ANNOTATIONS

We want to test if the non-expert annotations are reliable enough to train accurate SWSD classifiers and if we can exploit them for successful sense-aware subjectivity analysis.

4.3.1 In Vivo Evaluation

In section 4.2.2.2, we see that MTurk annotations has a very good agreement with the expert annotations. Now, we want to see if we can train accurate SWSD classifiers on them. For this purpose, we compare the performance of a SWSD system trained on non-expert annotations

	Acc	p-value
$\text{SWSD}_{\text{GOLD}}$	79.2	-
SWSD_{MJL}	78.4	0.542
SWSD_{MJC}	78.8	0.754

Table 13: Comparison of SWSD systems

with a SWSD system trained on expert annotations.

We use Group3 data. Note that we gathered three labels for each instance. This gives us two options to train the non-expert SWSD system: (1) training the system on the majority vote labels *($SWSD_{MJL}$)* (2) training three systems on the three separate label sets and taking the majority vote prediction *($SWSD_{MJC}$)*. Additionally, we train an expert SWSD system *($SWSD_{GOLD}$)* – a system trained on gold standard expert annotations. All these systems are trained on 40 instances of the eight target words for which we have both non-expert and expert annotations and are evaluated on the remaining instances of the gold-standard corpus. This makes a total of 923 test instances for the eight target words with a majority class baseline of 61.8.

Table 13 reports micro-average accuracy of each system and the two-tailed p-value between the expert SWSD system and the two non-expert SWSD systems. The p-value is calculated with McNemar's test. It shows that there is no statistically significant difference between classifiers trained on expert gold-standard annotations and non-expert annotations. These results provide evidence that non-expert annotations are as good as expert annotations for training SWSD classifiers. We adopt SWSD_{MJL} in all our following experiments, because it is more efficient.

4.3.2 In Vitro Evaluation

In section 4.3.1, we see that we can rely on non-expert annotations to train accurate SWSD classifiers. Now, we take a larger annotation effort and test if we can exploit non-expert SWSD for sense-aware subjectivity analysis. This section gives details on the conducted experiments to test if we can exploit non-expert SWSD for successful sense-aware subjectivity analysis and if learning based SWSD integration will perform better than rule-based SWSD integration

4.3.2.1 Data Annotation For our experiments, we have multiple goals, which effect our decisions on how to create the subjectivity sense-tagged corpus via MTurk. First, we want to be able to disambiguate more target words. This way, SWSD will be able to disambiguate a larger portion of the MPQA Corpus allowing us to evaluate the effect of SWSD on contextual opinion analysis on a larger scale. This will also allow us to investigate additional integration methods of SWSD into contextual opinion analysis rather than simple ad hoc manual rules. Second, we want to show that we can rely on non-expert annotations instead of expert annotations, which will make an annotation effort on a larger-scale both practical and feasible, timewise and costwise. Optimally, we could have annotated via MTurk senSWSD in order to compare the effect of a non-expert SWSD system on contextual opinion analysis directly with the results reported for an expert SWSD system. But, this would have diverted our resources to reproduce the same corpus and contradict our goal to extend the subjectivity sense-tagged corpus to new target words. Moreover, we have already shown in Section 4.3.1 that non-expert annotations can be utilized to train reliable SWSD classifiers. It is reasonable to believe that similar performance on the SWSD task will reflect to similar improvements on contextual opinion analysis. Thus, we decided to prioritize creating a subjectivity sense-tagged corpus for a totally new set of words. We aim to show that the

favourable results will still hold on new target words relying on non-expert annotations.

We chose our target words from the subjectivity lexicon of [Wilson et al., 2005], because we know they have subjective usages. The contextual opinion systems we want to improve rely on this lexicon. We call the words in the lexicon *subjectivity clues*. At this stage, we want to concentrate on the frequent and ambiguous subjectivity clues. We chose frequent ones, because they will have larger coverage in the MPQA Corpus. We chose ambiguous ones, because these clues are the ones that are most important for SWSD. Choosing most frequent and ambiguous subjectivity clues guarantees that we utilize our limited resources in the most efficient way. We judge a clue to be ambiguous if it appears more than 25% and less than 75% of the times in a subjective expression. We get these statistics by simply counting occurrences in the MPQA Corpus inside and outside of subjective expressions.

There are 680 subjectivity clues that appear in the MPQA Corpus and are ambiguous. Out of those, we selected the 90 most frequent that have to some extent distinct objective and subjective senses in WordNet. We annotated the WordNet senses of those 90 target words. For each target word, we selected approximately 120 instances randomly from the *GIGAWORD Corpus*. In a first phase, we collected three sets of MTurk annotations for the selected instances. In this phase, MTurk workers base their judgements on two sense sets they observe. This way, we get training data to build SWSD classifiers for these 90 target words.

The quality of these classifiers is important, because we will exploit them for contextual opinion analysis. Thus, we evaluate them first by 10-fold cross-validation. We split the target words into three groups. If the majority class baseline of a word is higher than 90%, it is considered as *skewed* (skewed words have a performance at least as good as the majority class baseline). If a target word improves over its majority class baseline by 25% in accuracy, it is considered as *good*. Otherwise, it is considered as *mediocre*. This way, we end up with

24 skewed, 35 good, and 31 mediocre words. There are many possible reasons for the less reliable performance for the mediocre group. We hypothesize that a major problem is the similarity between the objective and subjective sense sets of a word, thus leading to poor annotation quality. To check this, we calculate the agreement between three annotation sets and report averages. The agreement in the mediocre group is 78.68%, with a κ value of 0.57, whereas the average agreement in the good group is 87.51%, with a κ value of 0.75. These findings support our hypothesis. Thus, we created usage inventories for the words in the mediocre group. Usage inventories are basically lists of sample usages of a target word. We group them into two sets according their subjectivity as we do with senses, from which the worker can choose the most similar set. We initiated a second phase of MTurk annotations. We collect for the mediocre group another three sets of MTurk annotations for 120 instances, this time utilizing usage inventories. The 10-fold cross-validation experiments show that nine of the 31 words in the mediocre group shift to the good group. Only for these nine words, we accept the annotations collected via usage inventories. For all other words, we use the annotations collected via sense inventories. From now on, we will refer to this non-expert subjectivity sense-tagged corpus consisting of the tagged data for all 90 target words as the *MTurkSWSD Corpus* (agreement on the entire MTurkSWSD corpus is 85.54%, κ:0.71). These 90 target words have 3737 instances in the MPQA Corpus. We refer to this subset of the MPQA Corpus as *MTurkMPQA*. This subset makes up the coverage of the SWSD trained on the MTurkSWSD Corpus. Note that MTurkMPQA is 5.2 times larger than senMPQA.

4.3.2.2 Rule-Based SWSD Integration In this section, we train a SWSD system on MTurkSWSD and evaluate its effect on the two expression-level contextual classifiers introduced earlier via rule-based integration. The experiments are analogous to the ones

	Baseline		Acc	OF	SF
MTurkMPQA	52.4% (O)	$O_{S/O}$	67.1	68.9	65.0
		R1R2	**71.1**	72.7	69.2
senMPQA	63.1% (O)	$O_{S/O}$	75.4	65.4	80.9
		R1R2	81.3	75.9	84.8

Table 14: S/O classifier with and without SWSD.

in Section 3.4.3.3 and 3.4.3.4. The only difference is that we train the SWSD system on MTurkSWSD and evaluate its effect on MTurkMPQA.

We use the exact same rules and adopt the same confidence threshold. Table 14 holds the comparison of the original contextual classifier and the classifier with SWSD support on senMPQA and on MTurkMPQA. $O_{S/O}$ is the original S/O classifier; R1R2 is the system with SWSD support utilizing both rules.

In Table 14 we see that R1R2 achieves 4% percentage points improvement in accuracy over $O_{S/O}$ on MTurkMPQA. The improvement is statistically significant at the $p < .01$ level with McNemar's test. It is accompanied with improvements both in subjective F-measure (SF) and objective F-measure (OF). It is not possible to directly compare improvements on senMPQA and MTurkMPQA since they are different subsets of the MPQA Corpus. SWSD support brings 24% error reduction on senMPQA over the original S/O classifier. In comparison, on MTurkMPQA, the error reduction is 12%. We see that the improvements on the large MTurkMPQA set still hold, but not as strong as on senMPOA. This might be due to the brittleness of the rule-based integration.

Table 15 holds the comparison of the original N/P classifier with and without SWSD

	Baseline		Acc	NF	PF
MTurkMPQA	70.6% (P)	$O_{N/P}$	72.3	82.0	39.8
		R4	**74.5**	84.0	37.8
senMPQA	73.9% (P)	$O_{N/P}$	79.0	86.7	50.3
		R4	81.6	88.6	52.3

Table 15: N/P classifier with and without SWSD

support on senMPQA and on MTurkMPQA. $O_{N/P}$ is the original N/P classifier; R4 is the system with SWSD support utilizing rule R4. Since our main focus is not rule-based integration, we did not run the second step of the polarity classifier. We report the second step result below for the learning-based SWSD integration in section 4.3.2.3.

In Table 15, we see that R4 achieves 2.2 percentage points improvement in accuracy over $O_{N/P}$ on MTurkMPQA. The improvement is statistically significant at the $p < .01$ level with McNemar's test. It is accompanied with improvement only in objective F-measure (OF). SWSD support brings 12.4% error reduction on senMPQA. On MTurkMPQA, the error reduction is 8%. We see that the rule-based SWSD integration still improves both contextual classifiers on MTurkMPQA, but the gain is again not as large as on senMPQA.

4.3.2.3 Learning SWSD Integration Now that we can disambiguate a larger portion of the MPQA Corpus, we can investigate machine learning methods for SWSD integration to deal with the brittleness of the rule-based integration. We introduce two learning methods to apply SWSD to the contextual classifiers. For the learning methods, we rely on exactly the same information as the rule-based integration: (1) SWSD output, (2) the contextual classifier's output, (3) the contextual classifier's confidence, and (4) the presence of another

	Acc	OF	SF
$O_{S/O}$	67.1	68.9	65.0
R1R2	71.1	72.7	69.2
$EXT_{S/O}$	**80.0**	81.4	78.3
$MERGER_{S/O}$	78.2	80.3	75.5

Table 16: S/O classifier with learned SWSD integration

clue instance in the same expression. The rationale is the same as for the rule-based integration, namely to relate sense subjectivity and contextual subjectivity. The learning methods are as follows :

Method1 : In the first method, we extend the machine learning features of the underlying contextual classifiers by adding (1) and (4) from above. We evaluate the extended contextual classifiers on MTurkMPQA via 10-fold cross-validation. Tables 16 and 17 hold the comparison of Method1 ($EXT_{S/O}$, $EXT_{N/P}$) to the original contextual classifiers ($O_{S/O}$, $O_{N/P}$) and to the rule-based SWSD integration (R1R2, R4). We see substantial improvement for Method1. It achieves 39% error reduction over $O_{S/O}$ and 25% error reduction over $O_{N/P}$. For both classifiers, the improvement in accuracy over the rule-based integration is statistically significant at the $p < .01$ level with McNemar's test.

Method2 : This method defines a third classifier that accepts as input the contextual classifier's output and the SWSD output and predicts what the contextual classifier's output should have been. We can think of this third classifier as the learning counterpart of the manual rules from Section 4.3.2.2, since it actually learns when to flip the contextual classifier's output considering SWSD evidence. Specifically, this merger classifier relies on four

	Acc	NF	PF
$O_{N/P}$	72.3	82.0	39.8
R4	74.5	84.0	37.8
$EXT_{N/P}$	79.1	85.7	61.1
$MERGER_{N/P}$	**80.4**	86.7	62.8

Table 17: N/P classifier with learned SWSD integration

machine learning features (1), (2), (3), (4) from above (the exact same information used in rule-based integration). Because it is a supervised classifier, we need training data where we have clue instances with the corresponding contextual classifier and SWSD predictions and also the actual contextual label. Fortunately, we can use senMPQA for this purpose. We train our merger classifier on senMPQA (we get contextual classifier predictions via 10-fold cross-validation on the MPQA Corpus) and apply it to MTurkMPQA. We use an SVM classifier from the Weka package [Witten and Frank., 2005] with its default settings. Tables 16 and 17 hold the comparison of Method2 ($MERGER_{S/O}$, $MERGER_{N/P}$) to the original contextual classifiers (Oo/s, $O_{N/P}$) and the rule-based SWSD integration (R1R2, R4). It achieves 29% error reduction over $O_{S/O}$ and 29% error reduction over $O_{N/P}$. The improvement on the rule-based integration is statistically significant at the $p < .01$ level with McNemar's test. Method2 performs better (statistically significant at the $p < .05$ level) than Method1 for the N/P classifier but worse (statistically significant at the $p < .01$ level) for the S/O classifier.

We see that Method2 is the best method to improve the N/P classifier, which is the first step of the contextual polarity classifier. To assess the overall improvement in polarity

		Acc	NF	NgF	PsF
MTurkMPQA	$O_{Ps/Ng/N}$	72.1	83.0	34.2	15.0
	$MERGER_{N/P}$	**77.8**	87.4	53.0	27.7
senMPQA	$O_{Ps/Ng/N}$	77.6	87.2	39.5	40.0
	R4	80.6	89.1	43.2	44.0

Table 18: Polarity classifier with and without SWSD.

classification, we run the second step of the contextual polarity classifier after correcting the first step with Method2. Table 18 summarizes the improvement propagated to Ps/Ng/N classification. For comparison, we also include results on senMPQA. Method2 results in 20% error reduction in accuracy over $O_{Ps/Ng/N}$ (R4 achieves 13.4% error reduction on senMPQA). The improvement on the rule-based integration is statistically significant at the $p < .01$ level with McNemar's test. More importantly, the F-measure for all the labels improves. This indicates that non-expert MTurk annotations can replace expert annotations for our end-goal – improving contextual subjectivity analysis – while reducing time and cost requirements by a large margin. Moreover, we see that the improvements scale up to new subjectivity clues.

4.4 SUMMARY AND DISCUSSION

In this chapter, we utilized a large pool of non-expert annotators (MTurk) to collect subjectivity sense-tagged data for SWSD. We presented our subjectivity sense annotation task to MTurk workers in a very simple way. The annotation results show that subjectivity word sense annotation can be done reliably by MTurk workers. This is very promising since the

MTurk annotations can be collected for low costs in a short time period.

We showed that non-expert annotations are as good as expert annotations for training SWSD classifiers. The additional subjectivity sense-tagged data enabled us to evaluate the benefits of SWSD on contextual subjectivity analysis on a subset of MPQA that is five times larger than senMPQA. We demonstrated that SWSD classifiers trained on non-expert annotations can be exploited to improve contextual opinion analysis. The results support our fourth hypothesis:

Hypothesis 4: Crowdsourcing can be utilized to collect high-quality SWSD annotations in order to train SWSD classifiers with a good performance.

We also experimented with new learning strategies for integrating SWSD into contextual subjectivity analysis. With the learning strategies, we achieved greater benefits from SWSD than the rule-based integration strategies on all of the contextual subjectivity analysis tasks.

All these results imply that a large scale general SWSD component, which can help with various subjectivity and sentiment analysis tasks, is feasible. Overall, we more firmly demonstrated the potential of SWSD to improve contextual subjectivity analysis.

This chapter also contributes to ongoing work on crowdsourcing – to be specific MTurk – to create data for human language technologies . We addressed the question of whether built-in qualifications are enough to avoid spammers. The investigation of worker performances indicates that the lesser constrained a group is the more spammers it attracts. On the other hand, we did not find any significant difference between the quality of the annotations for each group. It turns out that workers considered as spammers contributed only minimally. We do not know if it is just a coincidence or if it is correlated to the task definition. We need to do more extensive experiments before arriving at conclusions.

Another aspect we investigated is the learning effect. Our results show that there is no

improvement in annotator reliability over time. We should not expect MTurk workers to provide more consistent annotations over time. This will probably be the case in similar annotation tasks. For harder annotation tasks (e.g. parse tree annotation) things may be different. An interesting follow-up would be whether showing the answers of other workers on the same HIT will promote learning.

Non-expert annotations acquired through MTurk can provide an alternative to expert annotations for many NLP tasks. Although non-expert annotations are inexpensive and fast, the collection process requires quality control mechanisms to ensure high-quality. In addition to built-in qualifications MTurk provides, the providers can implement voting schemes, check points and hidden gold units to screen out unreliable workers and improve quality. We think that an incremental approach is very useful. To be specific, providers should send data in subsequent iterations and let only reliable workers continue to the next iteration.

These mechanisms address only one side of the problem, namely unreliable workers. Perhaps, a more important point is the task definition and design. If a task is too complex and the instructions and design of a HIT are not clear, we cannot expect to collect reliable annotations even from well-meaning workers. It is very important to represent a task in a clear way with simple instructions. If the task is too complex or requires some amount of expert knowledge, it is best to simplify the task as we did in our experiments or to divide it in multiple less complex subtasks. [Negri et al., 2011] describes this as a divide and conquer method. The authors are able to collect large-scale high-quality annotations for a complex multilingual textual entailment task. They propose to split a complex problem into self-contained and easy to explain subtasks that are easy to execute without much NLP expertise and suitable for integration of a variety of control mechanisms discussed earlier. To summarize, we believe that MTurk can be utilized to collect reliable data for even complex NLP tasks and the success depends on the task design and quality mechanisms. There is a

trade-off, though. The effort put on simplifying a task and assuring quality might at some point become more time-consuming and expensive than collecting expert annotations.

4.5 RELATED WORK

There has been recently an increasing interest in Amazon Mechanical Turk [Callison-Burch and Dredze, 2010]. Many researchers have utilized MTurk as a source of non-expert natural language annotation to create labeled datasets. In [Mrozinski et al., 2008], MTurk workers are used to create a corpus of why-questions and corresponding answers on which QA systems may be developed. [Kaisser and Lowe, 2008] work on a similar task. They make use of MTurk workers to identify sentences in documents as answers and create a corpus of question-answer sentence pairs. [Parent and Eskenazi, 2010] produces a new sense-tagged corpus for WSD. MTurk is also considered in other fields than natural language processing. For example, [Sorokin and Forsyth, 2008] utilizes MTurk for image labeling and [Le et al., 2010] uses MTurk to collect handwritten text and their transcripts. Our ultimate goal is similar; namely, to build training data (in our case for SWSD).

Several studies have concentrated specifically on the quality aspect of the MTurk annotations. They investigated methods to assess annotation quality and to aggregate multiple noisy annotations for high reliability. [Snow et al., 2008] report MTurk annotation quality on various NLP tasks (e.g. WSD, Textual Entailment, Word Similarity) and define a bias correction method for non-expert annotators. [Callison-Burch, 2009] uses MTurk workers for manual evaluation of automatic translation quality and experiments with weighed voting to combine multiple annotations. [Negri et al., 2011] defines a data collection methodology on MTurk to ensure data quality. [Hsueh et al., 2009] define various annotation quality measures and show that they are useful for selecting annotations leading to more accurate

classifiers. Our work investigates the effect of built-in qualifications on the quality of MTurk annotations.

[Hsueh et al., 2009] applies MTurk to get sentiment annotations on political blog snippets. On a similar task, [Yano et al., 2010] applies MTurk to get political bias of blog snippets. [Snow et al., 2008] utilizes MTurk for affective text annotation task. In these works, MTurk workers annotated larger entities but on a more detailed scale than we do. [Snow et al., 2008] also investigates a WSD annotation task which is similar to our annotation task. The difference is the MTurk workers are choosing an exact sense not a sense set.

The strategy of presenting annotators with sets of usages rather than WordNet senses as we did for a sample of words was inspired by [Erk et al., 2009]. They carry out studies comparing, among other things, word sense judgments with respect to WordNet senses versus judgments of word usage similarities, and concluded that both tasks are well defined.

5.0 REDUCING ANNOTATION EFFORT: CLUSTER AND LABEL

In chapter 4, we see that MTurk can be utilized to collect high-quality annotations for SWSD and that non-expert SWSD can improve contextual subjectivity analysis. Although non-expert annotations are cheap and fast, they still incur some cost. In this chapter, we aim to reduce the human annotation effort needed to generate the same amount of subjectivity sense tagged data by using a small amount of labeled data and *context clustering*. We hypothesize that we can obtain large sets of labeled data by labelling clusters of instances.

We are inspired by how lexicographers create sense inventories. They collect occurrences of a word in a corpus and group different usages into coherent sets, which they later code as dictionary definitions. Our goal is similar. We represent each word instance as a feature vector (i.e. context vector) that describes its context. We try to group word instances into coherent clusters. If the clusters are reasonably pure – in terms of the meanings of the word instances they hold –, we can label clusters as a whole instead of labelling all the instances of a word separately. Of course, such an approach will introduce noise in the labeled data that we want to keep minimal. Thus, we experiment with novel techniques to achieve pure clusters: (1) improving the context representation with the help of compositional semantic models, (2) incorporating the notion of subjectivity into the context representation, and (3) utilizing constrained clustering to incorporate prior subjectivity knowledge into the clustering process. Part of the research presented in this chapter is published in [Akkaya et al., 2012]

In section 5.1, we introduce context clustering and distributional semantic models. Sec-

tion 5.2 describes proposed methods to obtain expressive contextual representations and their evaluation on the context clustering task. In Section 5.3, we give details on the "cluster and label" approach and describe our semi-supervised clustering algorithm. In the same section, we evaluate the quality of the semi-automatically generated subjectivity sense-tagged data.

5.1 CONTEXT CLUSTERING

The goal of *context clustering* is to cluster target word instances, so that the induced clusters contain instances used with the same sense. Context clustering takes as input a set of word instances represented as feature vectors – also called *context vectors*. The instances are clustered based on the similarity of their feature vectors. [Schutze, 1998] and [Purandare and Pedersen, 2004] are two prominent works in this field. The biggest difference between them is how they represent of the context of a word instance. [Schutze, 1998] uses a *distributional semantic model* (DSM) to create context vectors. In contrast, [Purandare and Pedersen, 2004] represents context vectors using local features common to supervised WSD (Table 7).

5.1.1 Distributional Semantic Models

Distributional semantic models (DSMs) [Turney and Pantel, 2010, Sahlgren, 2006, Bullinaria and Levy, 2007] provide a means for representing word meaning. They are based on the assumption that the meaning of a word can be inferred from its distribution in text.

A DSM is basically a co-occurrence matrix – also called *semantic space* – such that each row vector represents the distribution of a target word across contexts. The context can be a document, a sentence, or a word window around the target word. In this work, we focus on the latter one. In that setting, the dimensions of the vector represent co-occurring context

	computer	cheese	button	cat
$\overrightarrow{mouse}$	22	8	16	13
$\overrightarrow{click}$	23	0	18	0
$\overrightarrow{catch}$	0	2	0	11

Table 19: A hypothetical word-word co-occurrence matrix

words and hold some score based on the occurrence frequency of the context word near the target word in the specified window. This co-occurrence vector builds the semantic signature of the target word. Basically, each target word is described in terms of co-occurring words in its textual proximity. Table 19 represents a hypothetical word-word co-occurrence matrix (i.e. semantic space) for the words *"mouse"*, *"click"* and *"catch"*. The dimensions of the semantic space are *"computer"*, *"cheese"*, *"button"* and *"cat"*. The co-occurrence vector of the word "cat", $\overrightarrow{cat}$, is [22, 8, 17, 13], co-occurrence vector of the word "catch", $\overrightarrow{catch}$, is [0, 2, 0, 11] and co-occurrence vector for the word "click", $\overrightarrow{click}$, is [23, 0, 18, 0]. In this example, the matrix holds simple co-occurrence frequencies but it can be defined to have an association score between the target word and context words such as point-wise mutual information.

Note that DSMs model meanings of words out of context. This means that the rows of the co-occurrence matrix represent word types rather than word tokens. All contexts and senses of a target word are accumulated into one vector. For a token-based treatment, [Schutze, 1998] utilizes a second-order representation. That is, [Schutze, 1998] represents each target word token by averaging type vectors – rows of the semantic space – of the neighbouring words that occur in its context. For example, the word *mouse* in the sentence *"He caught*

the mouse" will be represented as $\frac{\overrightarrow{catch} + \overrightarrow{mouse}}{2} = [11\ 5\ 8\ 12]$.

5.2 COMPOSITIONAL MODELS

Compositional Models [Erk and Padó, 2008, Mitchell and Lapata, 2008, Thater et al., 2009] offer a powerful tool to represent words in context. They build on top of conventional DSMs. The meaning of a word in context (i.e., word token) is computed through composition operations applied to the target word and its context. [Mitchell and Lapata, 2008] evaluate a good amount of composition operations. Vector summation and element-wise vector multiplication are two sample composition operations from [Mitchell and Lapata, 2008]. To illustrate, Table 20 gives compositional vectors for the word "mouse" in the context of "click" and "catch" for both operations. *click+mouse* is computed by summing co-occurrence vectors of "click" and "mouse" from the semantic space in Table 19. *click·mouse* is computed by element-wise multiplication of the co-occurrence vectors of "click" and "mouse" from the same semantic space. The same is true for *catch+mouse* and *catch·mouse*. Note that the vectors in Table 19 and in Table 20 have the same dimensions. The difference is that the semantic space in Table19 is type-based, where the semantic space in Table 20 is token-based. In the token-based semantic space, "mouse" will have a different semantic vector depending on its context. From now on, we will refer to the co-occurrence vectors in the type-based semantic space as *type vectors* and the co-occurrence vectors in the token-based semantic space as *token vectors*.

From a linguistic perspective, it is appealing that the multiplicative model allows one vector to pick out the relevant content of the other. Indeed, [Mitchell and Lapata, 2008] show that element-wise multiplication performs overall better than vector addition and other

	computer	cheese	button	cat
$\overrightarrow{click} + \overrightarrow{mouse}$	45	8	35	13
$\overrightarrow{catch} + \overrightarrow{mouse}$	22	10	17	13
$\overrightarrow{click} * \overrightarrow{mouse}$	506	0	306	0
$\overrightarrow{catch} * \overrightarrow{mouse}$	0	16	0	143

Table 20: Additive and multiplicative composition of co-occurrence vectors

composition operations on a phrase similarity task without the need of parameter tuning. Thus, in our work, we rely on element-wise multiplication to derive contextual meaning of words, but there are some obstacles we need to address first. First of all, [Mitchell and Lapata, 2010] apply their models in a constrained setting applying it to word-pairs related with specific dependency relations (e.g. verb-object). It is not clear how to apply this model in longer context. Consider the example in Figure 13. Taking into account only the verb-object relation and computing *begin·strike* as the compositional meaning of "strike" is not sufficient in this context. The words that are most informative for disambiguating "strike" are "workers" and "mines". "workers" is related to strike over a "nsubj←|dobj→" dependency path and "mines" is connected to "strike" over a "prep–at←| nsubj←|dobj→" dependency path. That simple example shows that we need to utilize longer dependency paths to reach informative and discriminative context words. Thus, we propose to extend the model proposed in [Mitchell and Lapata, 2008] to include longer arbitrary dependency paths.

A simple strategy would be to utilize all possible context words connected to the target word through a dependency path to compute the compositional representation of the target

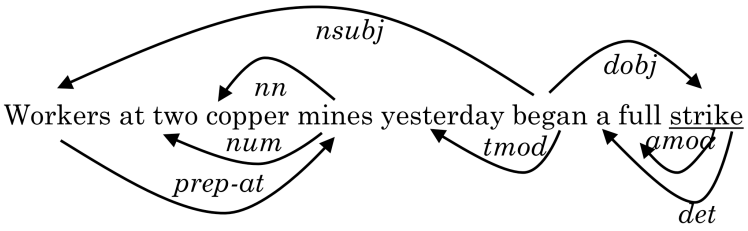

Figure 13: Example for compositional representation

word. But, that might introduce too much noise, since we can reach every word in the sentence if we fully traverse the dependency tree. Thus, we propose to investigate methods to filter out uninformative dependency paths (i.e. context words).

5.2.1 Exploiting Richer Contexts

In this section, we introduce the methods we use to choose informative context words of a target word to incorporate into the compositional representation of that target word.

From now on, we will refer to context words that are related to the target word over a dependency path as *context clues*. The most important question is how to filter out the uninformative context clues. We try four different methods for this purpose. The first two of them are simple in nature. They define constraints on the type of the context clue :

- *content* : the context clue should be a content word (i.e., noun, verb, adjective, or adverb).
- *nostop* : the content clue cannot be a stop word.

The next two are more elaborate filtering mechanisms. We define two scoring functions to assign an importance score to each context clue. Our intuition is that context clues which

carry more information to disambiguate the sense of the target word token should get a higher score and be chosen to contribute to the compositional representation of the target word (e.g. "workers" and "mines" in the figure 13 rather than "began").

The first scoring function keeps track of the change of the type vector of the target word after applying the type vector of the context clue to it. The hypothesis is that a context clue which selects out a specific sense of the target word will zero out a substantial amount of dimensions of the type vector of the target word (i.e., the more dimensions the context clue zeros out, the better the disambiguation should be). We count the dimensions of the type vector of the target word which become zero after applying the context clue. In order to avoid very infrequent context words getting high scores (since they will have lots of zero dimensions), we put a normalizing factor, the number of zero dimensions of the context clue. The scoring function, *maxzero*, is as follows: (*zero* is a function which returns the number of zero dimensions in a vector).

$$maxzero(target, clue) = \frac{zero(target) - zero(target \cdot clue)}{zero(clue)}$$

The second scoring function takes into account distributional substitutes of the target word based on the dependency path and the context clue. By distributional substitutes, we mean the set of words which are connected to the context clue via the same dependency relation in our corpus (described in Section 5.2.2.1). Our hypothesis is that the substitute sets can provide us useful information about the discriminative power of the corresponding context clues. If one of the substitutes related to a context clue is similar to one of the senses of the target word more strongly than other senses, we can conclude that the context clue is discriminative and should be scored high. For this purpose, we make use of WordNet similarity measure introduced by [Leacock and Chodorow, 1998]. First we assume that

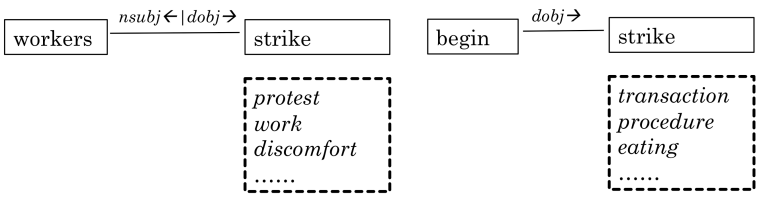

Figure 14: Example for distributional substitutes

each sense of a target word is equally probable and find the similarity of a substitute to different senses of the target word. Then, we normalize the similarity score over the senses and obtain a probability distribution over the senses of the target word. We apply the Kullback Leibler (KL) divergence to determine how much the new distribution differs from the uniform distribution. The context clues with substitution sets which have high maximum KL divergence scores are also scored high. Below is the formula for the *discsubs* scoring function. *subs* is a function which returns the set of distributional substitutes of a target word in context of a dependency path and context word. *disc* is a function which computes the KL divergence value as described above.

$$discsubs(target, clue, path) = \max_{s \in subs(target, clue, path)} disc(s, target)$$

$$disc(s, target) = D_{KL}(P(t_{senses}|s)||P(t_{senses})$$

To illustrate, in Figure 14, we see examples of distributional substitutes of strike for two context clues "workers" and "begin". One of the substitutes derived from the context clue "workers" is "protest", which is strongly related to the "work stoppage" meaning of

"strike" in WordNet. On the other hand, substitutes derived from the context clue "began" (e.g. "eating") are more general in nature and do not favour a specific sense of "strike" in WordNet. As a consequence, "workers" will get a higher score than "began", which is exactly what we want.

These two scoring functions give us two filtering mechanisms where we accept the best scoring context clues. We can choose more than one context clue to apply to the target word. By "applying" we mean element-wise multiplication of the type vector of the context clue with the type vector of the target word. We apply each chosen context clue separately to the target word resulting in multiple token vectors for the target word. We average these token vectors to obtain an ultimate single token vector of the target word. Our intuition is that each context clue chooses out some relevant dimensions of the target word and by averaging them, we smooth the contribution of the various context clues to create the final representation.

5.2.2 Experiments

This section gives details on the conducted experiments to evaluate the application of our extended compositional model for context clustering and compare it to existing popular context representations. In Section 5.2.2.1, we introduce the semantic space we rely on. Section 5.2.2.2 gives more detail on the context representations we experiment with. In Section, we measure the effect of using longer dependencies and filtering mechanisms on our development set 5.2.2.4 and compare all context representations for context clustering task on our test set in Section 5.2.2.5 to each other.

5.2.2.1 Semantic Space In this work, all approaches making use of a DSM – including our extended compositional model – use the same semantic space. The semantic space we use

in our experiments is built from a text corpus consisting of 120 billion tokens. We compile the corpus from various resources in order to have a balanced corpus. The corpus consists of news articles from GIGAWORD and editorials from NewYorker, NewYork Times, Slate, Townhall, BBC and Guardian. It also consists Open American National Corpus.

The rows of our semantic space correspond to word forms and the columns of the semantic space correspond to word lemmas present in the corpus. We adopt the parameters of our semantic space from [Mitchell and Lapata, 2010]: window size of 10 and dimension size of 2000 (i.e., the 2000 most frequent lemmas). [Mitchell and Lapata, 2010] found that setting to be optimal for a similar lexical semantic task. We do not filter out stop words, since they have been shown to be useful for various semantic similarity tasks in [Bullinaria and Levy, 2007]. We use positive point-wise mutual information to compute values of the vector components, which has also been shown to be favourable in [Bullinaria and Levy, 2007].

5.2.2.2 Context Representations [Schutze, 1998] represents each target word token by averaging type vectors of the neighbouring words that occur in its context. Note that it is similar to an additive model since all type vectors are added together. The model does not consider the syntactic dependencies in the context.

Our approach is similar to [Schutze, 1998], except that, instead of averaging *type vectors*, we average the *token vectors* of the neighbouring words that we compute with our extended compositional model. In our experiments we consider dependency paths of length up to four. For the *maxzero* and the *discsubs* filtering strategies, we need to specify how many context clues we want to choose. We try following variants : choosing highest ranking context clue, choosing two highest ranking context clues and choosing three highest ranking context clues. We also try a variant where we let all context clues contribute to the compositional representation of the target word, but we weight them by the inverse of their rank.

In contrast, [Purandare and Pedersen, 2004] utilize feature vectors similar to the ones common in supervised WSD. Specifically, we use the following features in Figure 15 from [Mihalcea, 2002b] to build the local feature representation. Note that we leave out global context features (i.e. *SK*), since they are extracted for each sense separately and require label information.

CW : the target word itself : nominal {1}
CP : the part of speech of the target word : nominal {1}
CF : the surrounding context of 3 words and their POS : nominal {12}
HNP : the head of the noun phrase to which the target word belongs : nominal {1}
NB : the first noun before the target word : nominal {1}
VB : the first verb before the target word : nominal {1}
NA : the first noun after the target word : nominal {1}
VB : the first verb before the target word : nominal {1}
VA : the first verb after the target word : nominal {1}

Figure 15: WSD features for SWSD

5.2.2.3 Clustering Algorithm and Evaluation Metric We use the same clustering algorithm for all context representations: agglomerative hierarchical clustering with average linkage criteria. In all our experiments throughout the paper, we fix the cluster size to 7 as it is done in [Purandare and Pedersen, 2004]. We think that is reasonable number since SENSEVAL III reports that the average number of senses per word is 6.47.

We choose *cluster purity* as our evaluation metric. To compute cluster purity, we assign each cluster to a sense label, which is the most frequent one in the cluster. The number of the correctly assigned instances divided by the number of all the clustered instances gives us cluster purity.

5.2.2.4 Effect of Longer Dependencies and Filtering Strategies Our first goal is to measure the effect of utilizing longer paths and the effect of the proposed filtering strategies. We want choose the best combination before comparing our extended model to other context representations. For this purpose, we did not want to use part of senSWSD as our development set, since we do not have many words in that set to begin with. We opted to use words from SENSEVAL II and SENSEVAL III that are not in senSWSD. Since we have ony fine-grained sense-tagged data for this set of words, we will evaluate our model for context clustering on fine-grained senses. This allows us to have a huge development set consisting of 96 words in total.

We use the same clustering algorithm for all context representations : *agglomerative hierarchical clustering with average linkage criteria*. We require 7 clusters as it is done in [Purandare and Pedersen, 2004]. We choose *cluster purity* as our evaluation metric. To compute cluster purity, we assign each cluster to a sense label, which is the most frequent one in the cluster. The number of the correctly assigned instances divided by the number of all the clustered instances gives us cluster purity. Following [Purandare and Pedersen, 2004], we assign each sense label to at most one cluster so that the assignment leads to a maximally accurate mapping of senses to clusters. The evaluation is done separately for each word.

The results are reported in Table 21. The rows are the dependency path lengths (e.g. L2 means we are using dependency paths of length at most 2) and the columns are the filtering strategies. For *maxzero* and *discsubs*, we have additional sub-columns that inform the number of highest scoring context clues we use. W means that we use a weighted average of all context clues based on their score.

The results show that using longer dependency paths can improve cluster purity. The best result is obtained when we consider dependency paths up to length 4 and utilize *maxzero* filtering strategy choosing only the highest scoring context clue. The results illustrate the

	nofilter	con	stop	maxzero				discsubs			
				1	2	3	W	1	2	3	W
L1	41.29	41.23	43.11	41.51	41.27	40.94	40.87	43.98	42.98	42.07	43.19
L2	39.34	41.18	43.01	44.75	41.70	40.68	40.57	44.28	42.12	42.26	42.49
L3	40.30	40.15	42.15	46.10	43.31	42.01	41.23	44.59	43.01	42.15	42.31
L4	40.30	40.26	42.23	**47.65**	44.49	42.19	41.07	44.66	43.10	43.80	43.12

Table 21: Effect of the various dependency path lengths and filtering techniques used to compute the contextual representation on the clustering performance

benefit of using longer dependency paths. Among the filtering strategies, *maxzero*, *discsubs* are consistently better than using no filtering, when we only use the highest scoring context clue. *nostop* also achieves better performance than no filtering. On the other hand, *content* is not better than using no filtering. If we do not utilize a filtering strategy and increase the path length, the purity suffers slightly. This provides evidence that filtering non-informative context clues is essential.

For comparison, we also evaluate the other context representations on our development set. The results are summarized in Table 23. *type_averaging* is the system based on [Schutze, 1998], *local_features* represents the system utilizing local feature representation and *token_averaging* in bold is our system relying on the extended compositional model using the best parameter setting – path length up to four with maxzero filtering considering only highest scoring context clue. The result show that our model improves over both representations for the context clustering task. The improvement is statistically significant at the $p < .05$ level based on a paired t-test. It is interesting to see that even without using longer

	Cluster Purity
token_averaging	**47.65**
type_averaging	39.01
local_features	41.85

Table 22: Comparison of context representations for context clustering on SENSEVAL

paths and multiplicative model does better than *type_averaging*.

5.2.2.5 Comparison of Context Representations In this section, we compare our extended model to previous models for the context clustering task on the coarse-grained senses. For this purpose, we use senSWSD dataset. It use all 39 words in this set. The results are summarized in Table 22. We use the same evaluation metric except that this time each sense label (e.g S and O) can be assigned to multiple clusters. The majority label baseline is 79.9.

Again, *type_averaging* is the system based on [Schutze, 1998], *local_features* represents the system utilizing local feature representation and *token_averaging* in bold is our system relying on the extended compositional model with the best parameter setting from Section 5.2.2.4. The result show that our model improves over both representations. The improvements are statistically significant at the $p < .05$ level based on a paired t-test. The results indicate that our extended model provides a better representation of the meaning of a word instance that we want to improve as much as possible for our end goal.

5.2.2.6 Merging Context Representations When we look at the context clustering results for single words separately on the development set, we observe that the performance

	Cluster Purity
token_averaging	**83.53**
type_averaging	80.49
local_features	80.49

Table 23: Comparison of context representations for context clustering on senSWSD

of different representations vary (Table 25). There is not a single winner among all words. *token_averaging* performs best for 59 of the words, *local_features* performs best for 36 of the words and *type_averaging* performs best for only 6 of the words. We want to get the best possible context representation. Thus, perhaps choosing one single representation for all the words is not optimal. Having that in mind, we try to merge *local_features* and *type_averaging*. We leave out *token_averaging*, since both *token_averaging* and *type_averaging* rely on the same type of semantic information (i.e. DSM). Moreover, *token_averaging* performs mostly worse than the other two. We believe that the two representations, *type_averaging* and *local_features*, one relying on a semantic space and the other one relying on local WSD features may complement each other.

We merge *local_features* and *type_averaging* to one single representation. We could have simply concatenated two feature vectors to one feature vector. But, there is an issue with this approach. The feature vectors of *local_features* and *type_averaging* have different scales and thus they will have different contributions to the final distance. To avoid this problem, we normalize each feature vector to unit length. This way, we make sure that the contribution of each underlying representation type is normalized. We call this method *mix_rep*.

In Table 24, we see that *mix_rep* performs better than all other three representation both on the development set (fine-grained) and test set (coarse-grained). The improvement

	Dev (SENSEVAL)	Test (senSWSD)
token_averaging	47.65	83.53
type_averaging	39.01	80.49
local_features	41.85	80.49
mix_context	**51.07**	**85.23**

Table 24: Effect of merging context representations

is statistically significant at the $p < .05$ level on both sets. When we look at the results for some sample words from the development set in Table 25, we observe that even if *mix_rep* does not perform always the best, it is never terrible either. It is consistently good and reliable. Thus, in later chapters *mix_rep* will be our choice as the context representation.

5.2.3 Incorporating Subjectivity into DSMs

Another extension we propose is modifying the underlying semantic space so that it mediates subjectivity. We hypothesize that building the subjective vs. objective distinction into the DSM will result in more discriminative context representation and thereby in purer context clusters in terms of subjectivity.

Distributional hypothesis dictates that words that occur in similar contexts tend to have similar meanings. Thus, the columns of the word-context matrix (i.e. dimensions of the semantic space) are essential for the similarity judgement. Using different set of dimensions will result in different similarity judgements and thus different clustering of word instances. We aim to modify dimensions of the underlying semantic space to incorporate subjectivity treatment. For this purpose, we experiment with two methods.

	type_averaging	*local_features*	*token_averaging*	*mix_rep*
activate-v	45.54	48.51	64.58	**79.76**
add-v	43.15	42.64	**53.05**	47.97
degree-n	43.08	**62.40**	56.14	**62.40**
dyke-n	37.28	55.93	55.93	**59.32**
provide-v	41.95	40.00	74.63	**79.02**
rule-v	37.08	44.94	**51.68**	47.19
wander-v	57.33	62.00	62.00	**78.00**

Table 25: Comparison of context representations for context clustering on SENSEVAL on sample words

In the first method, we want to use lexicon clues as dimensions of the semantic space, since subjectivity clues are associated with subjective language. We also consider *intensifiers* and *valence shifters* as dimensions of the semantic space. An intensifier is a word that has little semantic content of its own but that serves to intensify the meaning of the word or phrase that it modifies (e.g. "awfully" in the phrase "awfully sorry"). A valence shifters is a word that reverses the polarity of the phrase it modifies (e.g., little truth, little threat). Although intensifiers and valence shifters do not have subjectivity they are good clues that subjectivity has been expressed.

In the second method, we try to choose the dimensions of a semantic space based on their discriminative power between subjective and objective context of a target word. This means the dimensions of the semantic space will be tailored for the target word itself. We will have a different semantic space for each target word and use it for its context representation. The

important question is how do we find discriminative context words of a target word. We basically need some annotated instances of the target word in order to disambiguate between its subjective and objective context. Unfortunately, we do not have that information, since that is what we are trying to generate. [Riloff and Wiebe, 2003] uses a high-precision rule-based classifier (Section 3.4.3.2) to train a sentence-level subjectivity classifier. We use the same approach. We accept the sentence subjectivity as a signal for the word being subjective. We assume the probability of a word being is used with a subjective sense is higher if it occurs in a subjective sentence. We tag the sentences in our text corpus (Section 5.2.2.1) with the rule-based classifier. This gives us two smaller corpora, one subjective and the other one objective. For each target word, we find the context words in a window size of 10 in each corpus. Then, we compute *Pointwise Mutual Information* (PMI) between the target word and its context words in each corpus separately, PMI_{subj} and PMI_{obj}. Our idea is that context words that have a large difference between PMI_{subj} and PMI_{obj} are discriminative and should be chosen as dimensions of the semantic space. For this purpose we score the context words by $|PMI_{subj} - PMI_{obj}|$ score and choose highest ranking context word as dimensions. We consider only the context words that appear at least 300 times in our corpus in order to avoid that very infrequent words are chosen as dimensions

	Cluster Purity
Orig	**83.53**
Method1	82.68
Method2	83.11

Table 26: Effect of DSM modification

We evaluate the effect of both methods on context clustering task. For this purpose, we create our extended compositional model based on DSMs whose dimensions are chosen as described. We use the evaluation setting as in Section 5.2.2.5. Table 26 summarizes context clustering results on our test set. *Orig* is the original semantic space we use 5.2.2.1. *Method1* is the method where we choose lexicon clues, intensifiers and valenceshifters as dimensions. *Method2* is the method where we choose the dimensions based on their discriminative power. Note that for all three variants, the dimensionality of the semantic space is 2000. We see that both methods to incorporate subjectivity into the semantic space do not improve over the original semantic space. The results are very similar no matter which method we choose to create the dimensions of the semantic space.

5.3 LABELING CLUSTERS

Our ultimate goal is to reduce human effort to create training data for SWSD. We want to accomplish that using context clustering in a semi-automatic way. This annotation process has following steps:

- Cluster context vectors of word instances
- Label the induced clusters as S or O.
- Propagate the given label to all the instances in a cluster.

In this chapter, until now, we introduced our methods to improve the underlying context representation. Our methods result in a more informative context representation and thus in purer clusters, which will directly effect the quality of the semi-automatically generated SWSD data. There is an important question remaining. **How do we label clusters?** For a practical use of the *"cluster and label"* strategy, we need a way to label the clusters.

A straightforward strategy is to sample some instances from a cluster and label them. Then, we accept the majority label in the cluster as its label and propagate the chosen label back to all the instances in the cluster. This means we need to label some instances, preferably a small amount.

Since we will need a small amount of labeled data, we propose to use *semi-supervised clustering* to build the clusters. If we label some instances prior to clustering and we can use them to incorporate prior subjectivity knowledge into the clustering process. The provided labels will guide the clustering algorithm to generate the clusters that are more suitable for our end task, namely clusters where subjective and objective instances are grouped together. We do not have such an option if we utilize unsupervised clustering. Again, after the clusters are generated, we can propagate the majority label in a cluster back to all the instances in the cluster.

5.3.1 Constrained Clustering

Constrained clustering [Grira et al., 2004] also known as semi-supervised clustering is a recent development in the clustering literature. In contrast to unsupervised clustering, constrained clustering requires pairwise constraints. There are basically two types of constraints: (1) must-link and (2) cannot-link constraints. A must-link constraint dictates that two instances should be in the same cluster and a cannot-link dictates that two instances should not be in the same cluster. These constraints can be hard or soft. Hard constraints are those that we want definitely hold. In contrast, soft constraints do not have to be satisfied strictly. The constraints act as a guide for the clustering algorithm that will attempt to find clusters that satisfy the specified must-link and cannot-link constraints. The constraints can be obtained from domain knowledge or from available instance labels. There are generally two different strategies to incorporate constraints into the clustering. First strategy is to adapt

the underlying distance metric [Xing et al., 2002, Klein et al., 2002]. Second strategy is modifying the clustering algorithm itself so that search is biased towards a partitioning for which the constraints hold [Wagstaff and Cardie, 2000, Basu et al., 2002, Demiriz et al., 1999].

5.3.2 Iterative Constrained Clustering

Previous work report substantial improvement in the clustering accuracy with the usage of instance-level constraints. But, it is very important how to choose the constraints. [Davidson et al., 2006] show that even if the constraints are generated from gold-standard data, it is very common that some constraint sets can decrease clustering accuracy. The results are reported on the UCI datasets, which are not as hard as SWSD. Considering the difficulty of the SWSD task, choosing a good set of constraints becomes more important. Thus, we would like to choose constraints which we believe will have maximum impact on the clustering accuracy. For this purpose, we define a novel active selection strategy for constrained clustering. In order to choose most helpful constraints, we borrow ideas from active learning for classification. We call our algorithm *iterative constrained clustering* (ICC). As its name tells, we utilize an iterative process to create constraints actively. In each iteration the algorithm queries the most informative instance and acquires its label. The constraints are derived from the labels. Note that n labels result in $\binom{n}{2}$ constraints. An important question is how we define informativeness of an instance for clustering.

5.3.2.1 Informativeness We consider an instance to be informative if there is a high probability that the knowledge of its label will change the cluster boundaries. The more probable that change is, the more informative the instance is. Our basic idea is that if an instance is in a cluster holding instances of type a and it is close to another cluster holding instances of type b, that instance is most likely mis-clustered. Thus, it should be queried.

For this purpose, we define a scoring function, which is used to score each data point on its goodness, the lower the score the likely it is that the instance is mis-clustered. Choosing the data point with the lowest score, will likely change clustering borders in the next iteration. Our scoring function is based on silhouette coefficient. Silhouette coefficient is a popular unsupervised cluster validation metric to measure goodness [Tan et al., 2005] of a cluster member. It gives a score between -1 and 1. A higher score is better. [Tan et al., 2005] defines it as follows:

- for an instance i, compute its average distance from the other instances in its cluster x_i
- for an instance i, compute its average distance from the clusters in which the instance is not present and take the minimum of these averages y_i. Note that the average distance of an instance to a cluster is the average distance to all members of that cluster.
- compute the silhouette coefficient as $(y_i\text{-}x_i)$ / $\max(y_i,x_i)$

Basically, silhouette score assigns a cluster member that is close to another cluster a lower score and a cluster member that is closer to the cluster center a higher score. That is partly what we want. In addition, we do not want to penalize a cluster member that is close to another cluster having members with the same label. For this purpose, we calculate silhouette score only over clusters with an opposing label (i.e. holding members with an opposing label). In addition, we consider only so far labeled instances when computing the score. We call this new coefficient silh_{const}. It is computed as follows:

- for an instance i, compute its average distance from the other instances in its cluster x_i which are already labeled
- for an instance i, compute its average distance from the labeled instances of the clusters from an opposing label and take the minimum of these averages y_i
- compute silh_{inst} as $(y_i\text{-}x_i)$ / $\max(y_i,x_i)$

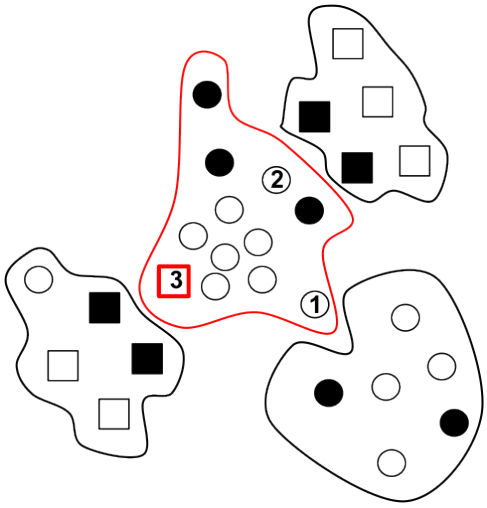

Figure 16: Behaviour of selection function

The silh$_{const}$ coefficient has favourable properties. First of all, it will score members that are close to a cluster with an opposing label lower than the members that are close to a cluster with the same label. According to our definition, these members are more informative. Figure 16 holds a sample cluster setting. The shape of a member denotes its label and filling denotes that it is has been queried. In this example, silh$_{const}$ will score members 2 and 3 lower than 1. Thus, member 1 will not be selected, which is the right decision in this example. Both members 2 and 3 are close to clusters with an opposing label. In this example silh$_{const}$ scores member 3 lower, which is farther away from already labeled members in the cluster. Thus, member 3 will be selected to be labeled. This type of behaviour results in an explorative strategy.

5.3.2.2 Imposing Constraints Our hypothesis is that in each iteration the algorithm will choose the most problematic instance, which will end up changing cluster boundaries. In order for that to happen, we need a mechanism to impose constraints. For this pur-

pose, we use distance metric learning similar to [Xing et al., 2002]. We use the method described in [Davis et al., 2007] to learn a new metric after each iteration. [Davis et al., 2007] presents an information-theoretic approach to learning a Mahalanobis distance function. The authors formulate the problem as minimizing the differential relative entropy between two multivariate Gaussians under constraints. The reason we choose the distance metric learning function [Davis et al., 2007] over [Xing et al., 2002] is that we believe it is more scalable.

In each iteration, the learned metric helps to rearrange the instances from opposing labels so that they are more distant from each other and the cluster boundaries are morphed. There is an issue though. A learned metric does not enforce that labels from opposing labels should not be assigned the same cluster. We can consider them as imposing soft constraints. But, we need these hard constraints. This means that the cluster should hold instances from one type, since our selection strategy requires it and our goal is to propagate a unique label to the unlabeled members of the cluster. In order to impose hard cannot-link constraints, we implement the mechanism by [Klein et al., 2002]. We set the distance between two cannot-linked instances to the maximum distance in the dataset and use agglomerative hierarchical clustering with complete-linkage. Complete-linkage step imposes hard cannot-link constraints.

5.3.2.3 Complete Algorithm ICC starts by simply clustering the instances without any constraints. The algorithms asks for labels of the prototypical members of each cluster. Then, the algorithm derives constraints from the labels and then the iterations begin. Algorithm 1 contains specific steps. For our problem, we only consider cannot-links, because of the definition of our SWSD task. For example, two instances can be labeled as subj, but that does mean that they should be similar to each other. They can be totally different usages

Algorithm 1 Iterative Constrained Clustering

X ... target clue instances

T = cluster(X)

L = labelprototypes(T)

while queries left **do**

  C = createconstraints(L)

  X = learnmetric(X,C)

  T = clusterwithhardconstraints(X,C)

  L = labelmostinformative(L,T)

end while

L = propagatelabels(L,T)

having subjective meaning. On the other hand, if two instances are labeled having opposing labels, we do not want them to be in the same cluster, since they are different usages. Thus, we only make use of cannot-link constraints.

5.3.3 Experiments

This section gives details on the conducted experiments to evaluate the purity of the semi-automatically generated subjectivity sense tagged data by our "cluster and label" strategy. We carry out detailed analysis to quantify the effect of metric learning (e.g. soft constraints) and proposed active selection strategy on the purity of the generated data and compare it to competitive baselines.

5.3.3.1 Compared Methods We implement the constrained clustering algorithm described in [Klein et al., 2002] as a baseline. Their algorithm operates on the distance matrix

between instances. It can handle both must-links and cannot-links. It imposes constraints by changing the distance matrix according the given constraints. Basically, the distances between must-linked instances are set to 0. That is not enough by itself, since if a is must-linked to b, instances close to a should become closer to b and also instances close to b should get closer to a. There is a need to propagate the constraint. This is done by calculating shortest path between all the instances and updating the distance matrix accordingly. To impose cannot-links, the distance between two cannot-linked instances is set to some large number. Complete-linkage step indirectly propagates the cannot-link constraints. To our knowledge, there have been only two previous work selecting constraints for constrained clustering actively [Basu et al., 2004, Klein et al., 2002]. The method described in [Basu et al., 2004] uses the farthest-first traversal scheme for informative selection of pairwise constraints. That strategy is not suitable for our setting, since we have only two labels. After sampling just one instance from both labels, this method becomes the same as random selection of constraints. The method described in [Klein et al., 2002] is simple. At first, the hierarchical clustering algorithm follows in a unconstrained fashion until some moderate number of clusters are remaining. Then the algorithm starts to request constraints between roots whenever two clusters are merged. We change the method slightly and provide labels of the roots instead of constraint between them. Since we have a binary task querying labels makes more sense than providing single constraints.

Our ICC method is closely related to the constrained clustering method described in [Klein et al., 2002]. We share the same backbone: the exact same hierarchical clustering algorithm and mechanism to impose hard constraints. There are two differences. We utilize a different active selection method, which makes our algorithm iterative, where [Klein et al., 2002] is a single pass algorithm. Second difference is that we have the capability of imposing soft constraints via metric learning.

In all the experiments, our proposed method and the baselines have the same values for the shared parameters. We utilize the *mix_rep* context representation. We require 7 clusters as in previous sections. We use *accuracy* as our evaluation metric. We assign a label to each cluster based on the labeled instances it holds. A cluster holding instances labeled as S will be labeled as S and vice versa. Then, the label is propagated to all of its instances. After this, it is straightforward to compute accuracy. This evaluation setting reflects a real-world scenario where we actually utilize ICC to generate data for SWSD. In our experiments, we only consider cannot-links, since even two instances are labeled with the same sense (S/O), the usages may be so different that forcing them in the same cluster will have negative effect on the clustering quality. Note that, we create more than 2 clusters for each target word.

5.3.3.2 Effect of Active Selection Strategy In this section, we evaluate ICC on senSWSD dataset. To be specific, we use SENSEVAL II and SENSEVAL III subsets of senSWSD, since we used SENSEVAL I subset as a development set while working on our active selection strategy. We report the accuracy of the semi-automatically generated data for different percentages of the queried data (e.g. 10% means that the algorithm queried 10% of the data to create constraints). This way, we obtain a learning curve. We report percentages, since the words in our test dataset have different number of instances.

Figure 17 holds the comparison of ICC with silh_{const} selection to a random selection baseline. "majority" stands for majority label frequency in the word set. We are interested in how well the system would perform on more and less ambiguous words. Thus, we split the words into three subsets according to their majority-class baselines – [50%,70%) , [70%,90%), and [90%,100%). We see that silh_{const} performs better than the random selection for all subsets of words. By providing labels to only 25% of the data, we can achieve 87.67% accurate fully labeled data.

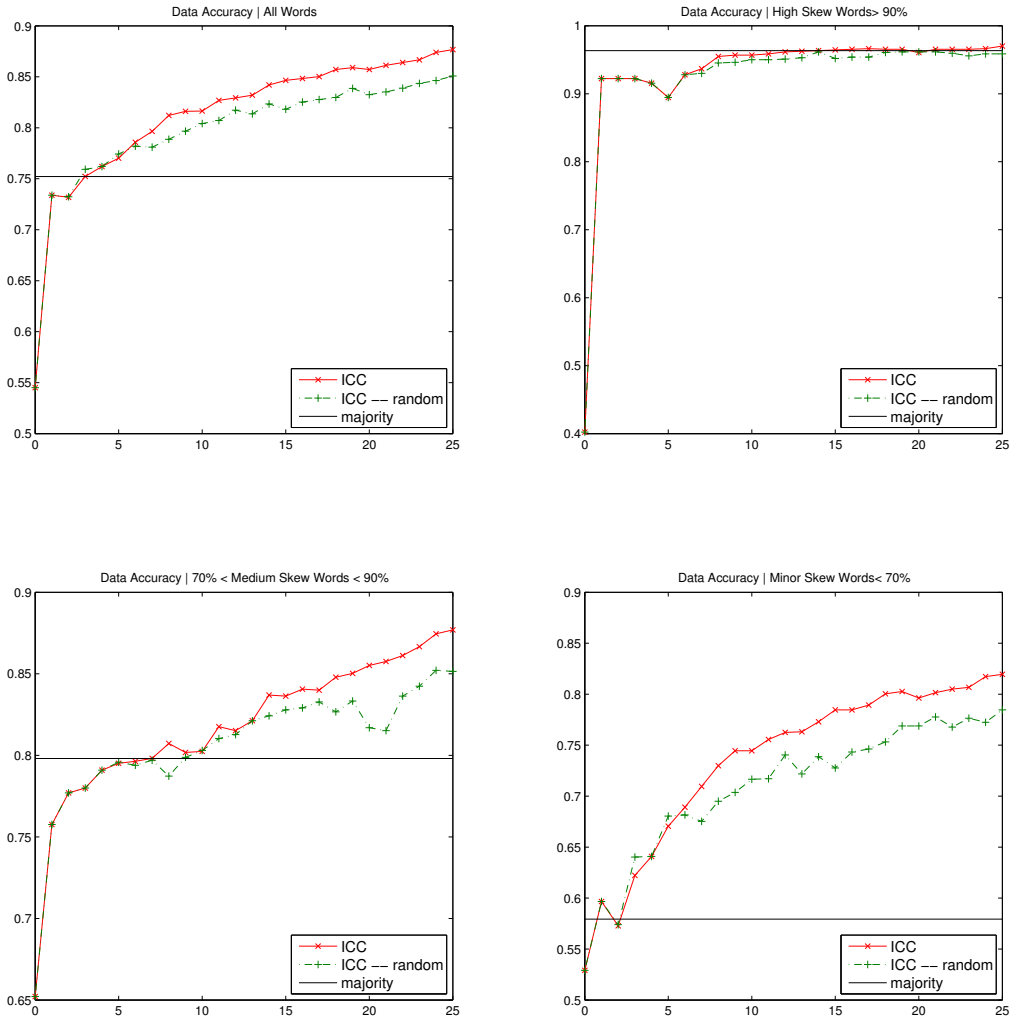

Figure 17: Accuracy of generated subjectivity sense tagged data – ICC vs. random selection

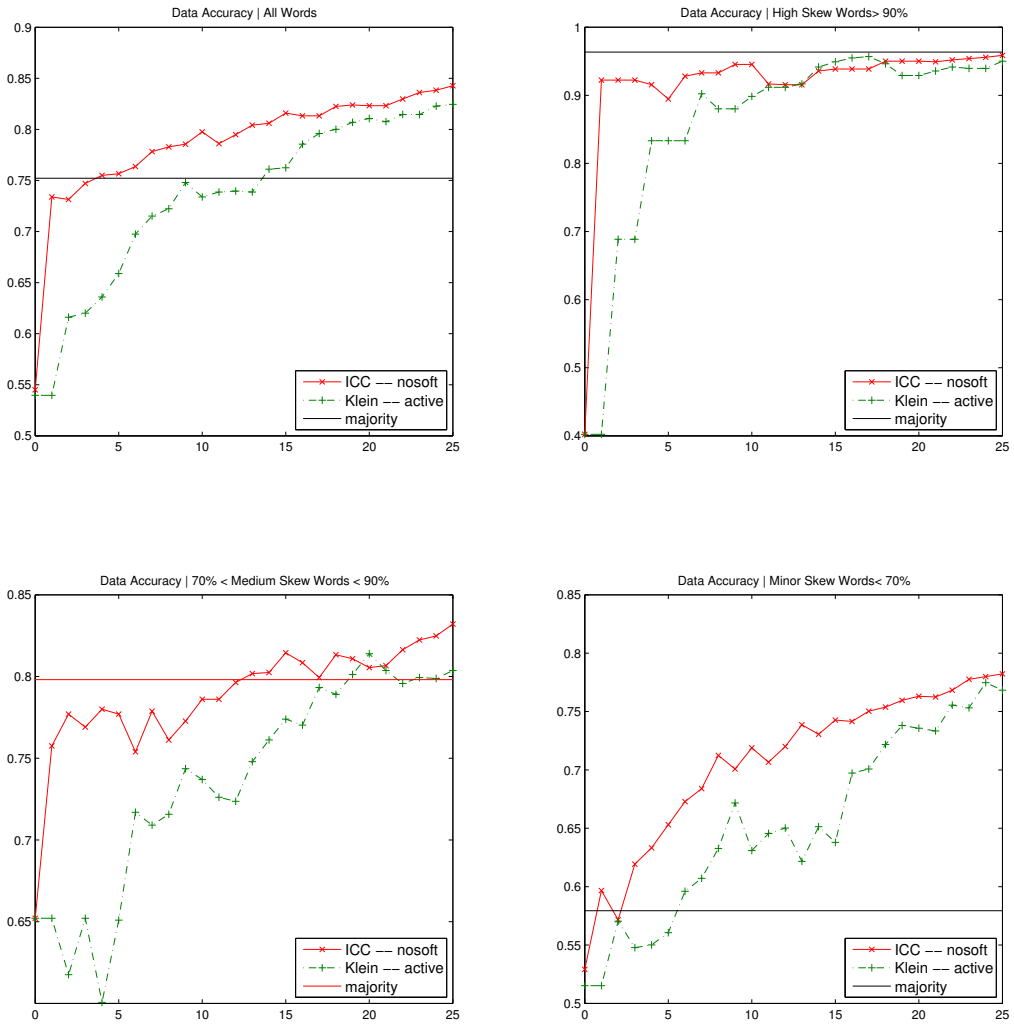

Figure 18: Accuracy of generated subjectivity sense tagged data – ICC without soft-contraints vs. Klein

For comparison, we also evaluate the performance of [Klein et al., 2002] with their active constraint selection strategy as described before. [Klein et al., 2002] does not use any soft constraints. Thus, we run our algorithm without soft constraints, in order to be able to compare the effectiveness of both active selection strategies. In Figure 18, we see that silh_{const} performs better than the active selection strategy described in [Klein et al., 2002] for all subsest of words.

5.3.3.3 Effect of Metric Learning We also wanted to investigate the effect of using soft-constraints via metric learning. For this purpose, we run our algorithm with and without metric learning. Figure 19 holds the results. For comparison, we also include [Klein et al., 2002]. We see that soft-constraints results in a big improvement. In addition, metric learning results in a smoother learning curve. That is a favourable property for a real-world application.

If we refer to Figure 17, we will see that our algorithm with metric learning even with random selection does better than both algorithms without metric learning.

5.3.3.4 Effect of Oracle Cluster Assignment For evaluation, we assign a label to each cluster based on the labeled instances it holds. The label is propagated to all of its instances and then accuracy is computed. Labelling clusters based on the instances they hold might introduce some error. If we had a human in the loop who can examine the clusters and assign the correct label to it, we might avoid these errors. In this section, we aim to answer the question "if we had a way to correctly label the clusters, what accuracy could we achieve?". For this purpose, we run our algorithm on senSWSD again, but this time we simulate an oracle who assigns a cluster the label that most of the members in the cluster share. Figure 20 holds the comparison of oracle evaluation to our original evaluation. We

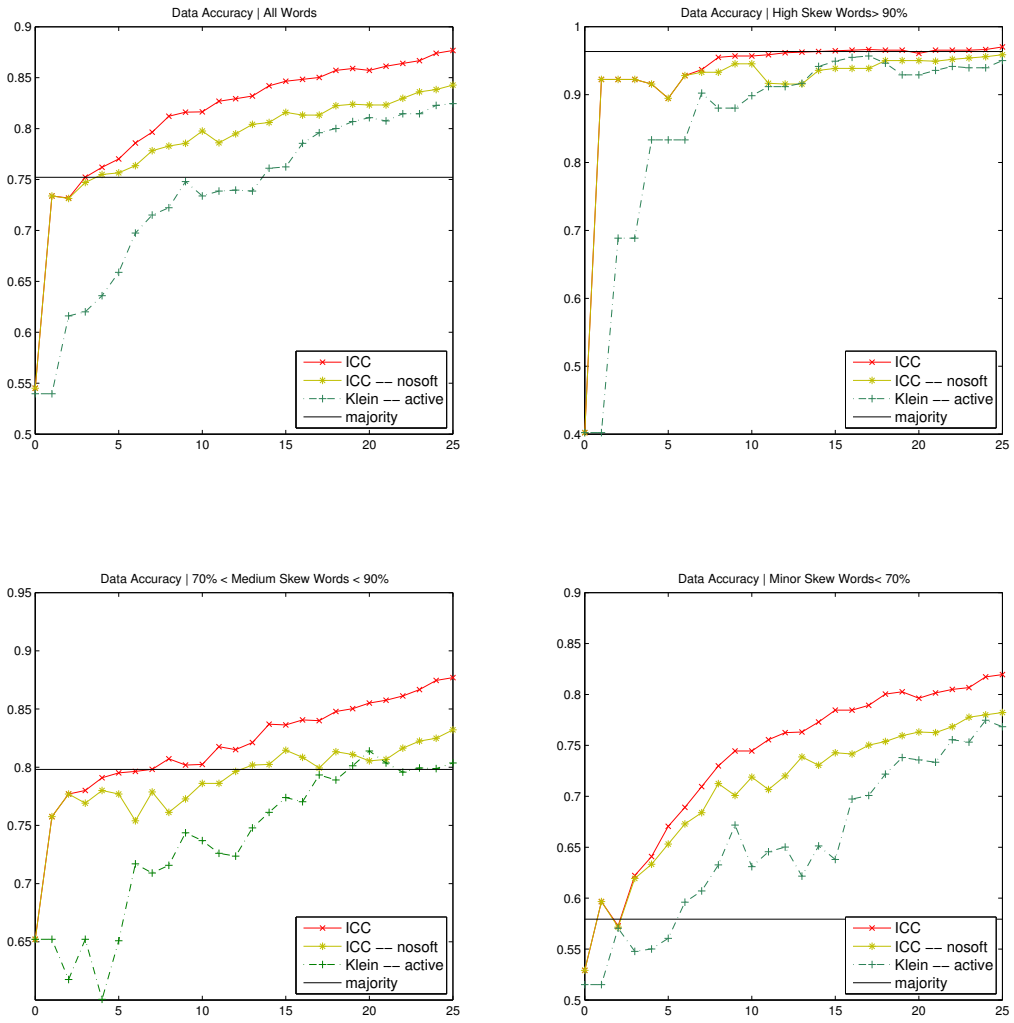

Figure 19: Accuracy of semi-automatically created data by ICC with and without soft-constraints

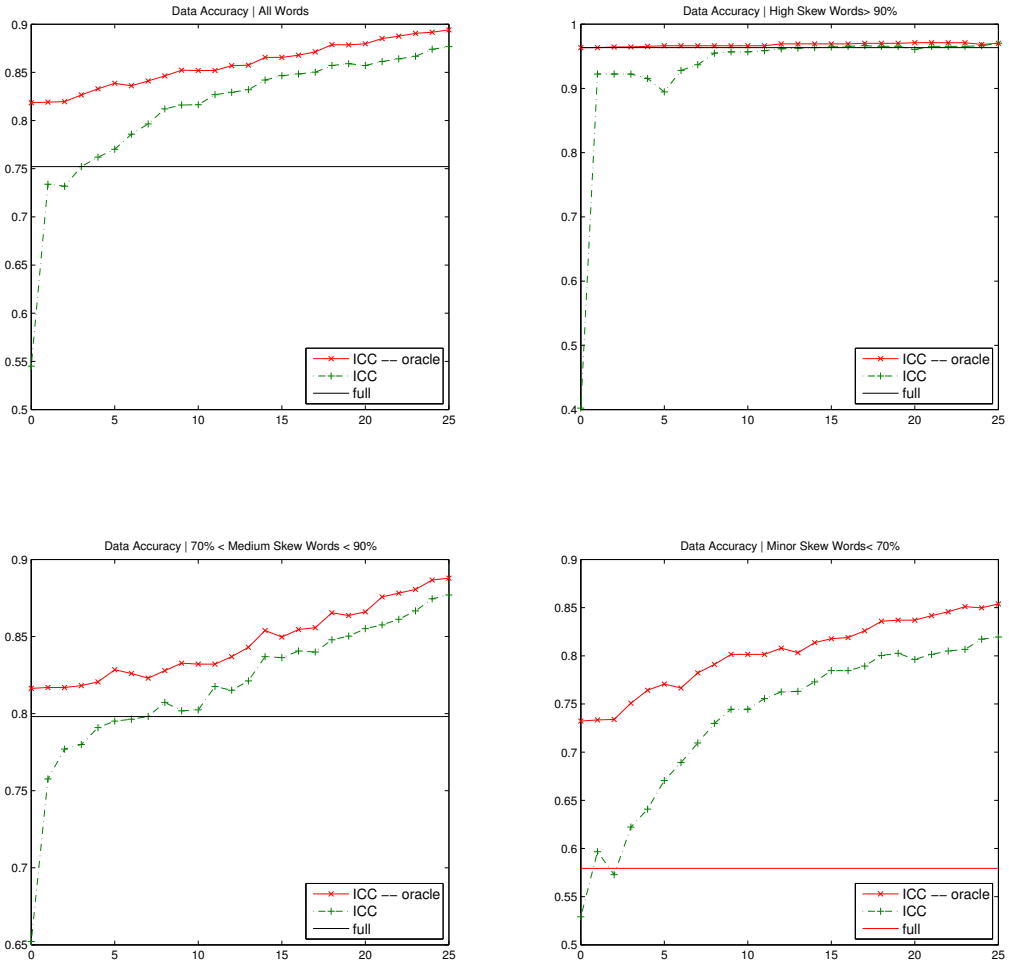

Figure 20: Accuracy of semi-automatically created data by ICC with oracle cluster assignment

see that we lose a fair amount of accuracy because of the errors we make while labelling the clusters. This error gets smaller over the iterations, since we get more labeled data and better evidence to judge the label of a cluster based on the labeled instances it holds. By providing labels to 25% of the data, ICC with oracle cluster assignment can achieve 89.42% accurate fully labeled data, which was 87.67 when we assign cluster labels based on the labeled instances they contain.

5.3.3.5 SWSD on semi-automatically generated annotations Now that we have a tool to generate training data for SWSD, we want to evaluate it on the actual SWSD task. We want to see if the obtained purity is enough to create reliable SWSD classifiers. In this section, we conduct our experiments on the MTurkSWSD dataset. There are two reasons for that. First one is that the MTurkSWSD dataset is more balanced in terms of the number of instances each word has. Second, it will allow us to see if we can combine two approaches – MTurk and ICC – to reduce annotation time and cost. Note that we do not use oracle cluster assignment in these experiments.

We conduct for each word in our dataset 10-fold cross-validation experiments. In each iteration, we apply ICC to training folds and label the instances semi-automatically. We train SWSD classifiers on the semi-automatically labeled training fold labels and test the classifiers on the corresponding test fold. When we train our classifiers we distinguish between queried instances and propagated labels. We weight the instances with propagated labels by their silh_{const} score, since that measure gives the goodness of an instance. The score is defined between -1 and 1. We normalize this score between 0 and 1, before using it as a weight. As our classifiers, we use the SVM classifier from the Weka package [Witten and Frank., 2005] with its default settings.

We implement two baselines. First one is simple *random sampling* and second one is

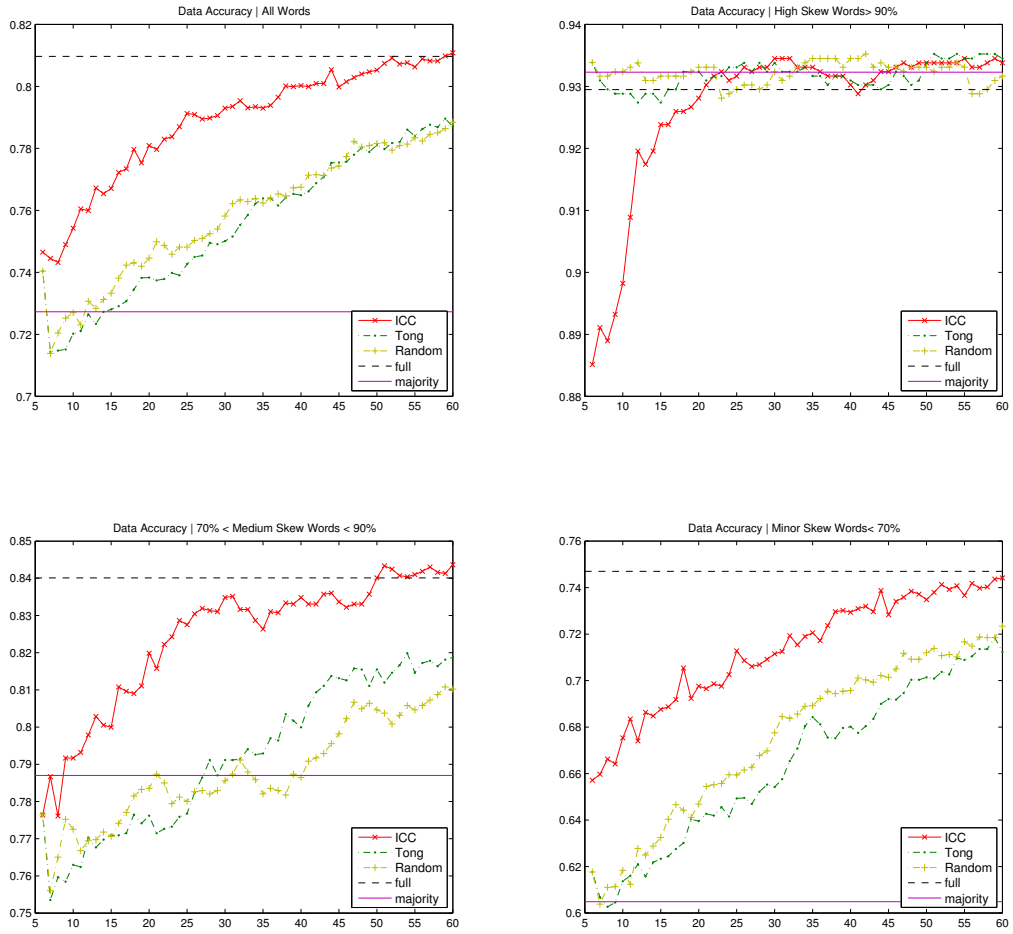

Figure 21: Accuracy of semi-automatically created data by ICC and baselines

Accuracy	75	76	77	78	79	80	80.5	80.98
ICC	10%	11%	16%	20%	25%	38%	44%	59%
Random	26%	31%	41%	47%	64%	72%	74%	100%
Tong	30%	34%	43%	48%	64%	73%	84%	92%
Reduction	62%	65%	61%	57%	61%	47%	41%	36%

Table 27: Annotation Reduction with ICC over Uncertainty and Random Sampling

uncertainty sampling, which is an *active learning* (AL) method. We use "simple margin" selection described in [Tong and Koller, 2001]. Simple margin technique selects in each iteration the instance closest to the decision boundary of the trained SVM. We run each method until it reaches the accuracy of training fully on the gold-standard data. ICC reaches that boundary when provided only 59% of the labels in the dataset. For uncertainty sampling and random sampling these values are 92% and 100% respectively. In Figure 21, we see the SWSD accuracy for different queried data percentages. "full" stands for training fully on gold-standard data. We see that training SWSD on semi-automatically labeled data by ICC does consistently better than uncertainty sampling and random sampling.

Table 27 holds a summary of the learning curves in Figure 21 for various accuracy points. We also report the annotation reduction we achieve over the best of the two baselines. It is surprising to see that uncertainty sampling overall does not do better than random sampling. We believe that it might be because of sampling bias. During AL, as more and more labels are obtained, the training set quickly diverges from the underlying data distribution. [Schütze et al., 2006] states that AL can explore the feature space in such a biased way that it can end up ignoring entire clusters of unlabeled instances. We think that SWSD is highly

prone for the mentioned missed cluster problem because of its unique nature. As mentioned, SWSD is a binary task where we distinguish between subjective and objective usages of a subjectivity word. Although the classification is binary, the underlying usages are grouped into multiple clusters corresponding to senses of the word. It is possible that two groups of usages represented very differently in the feature space are both subjective or objective. Moreover, one usage group might be closer to a usage group from opposing label than to a group with the same label.

We see that our method reduces the annotation amount by 36% in comparison to uncertainty sampling and by 41% in comparison to random sampling to reach the performance of the SWSD system trained on fully annotated data. As a last step, we want to see if at this threshold the improvements on the contextual classifiers still hold. We apply SWSD trained on semi-automatically generated training data to contextual S/O classifier and to the first step (N/P classifier) of the contextual polarity classifier. In Tables 29 and 28, we see that by labeling only 59% percent of the data we can achieve almost the same results.

	Acc
SWSD_{full}	80.0
$\text{SWSD}_{ICC-59\%}$	79.7
$\text{SWSD}_{AL-92\%}$	79.9

Table 28: S/O classifier with SWSD trained on semi-automatically generated annotations

	Acc
SWSD_{full}	80.4
$\text{SWSD}_{ICC-59\%}$	80.2
$\text{SWSD}_{AL-92\%}$	80.2

Table 29: N/P classifier with SWSD trained on semi-automatically generated annotations

5.4 SUMMARY AND DISCUSSION

In this chapter, we explore a "cluster and label" strategy to reduce the human annotation effort needed to generate subjectivity sense-tagged data. The basic idea is to label clusters of instances as a whole instead of labelling the instances of a word separately. In order to keep the noise in the semi-automatically labeled data minimal, we experiment with novel techniques to improve cluster purity by (1) improving the context representation with the help of compositional semantic models, (2) incorporating the notion of subjectivity into the context representation, and (3) utilizing constrained clustering to incorporate prior subjectivity knowledge into the clustering process.

We extended element-wise multiplication model introduced in [Mitchell and Lapata, 2008] to effectively incorporate richer contexts. Our experiments showed that longer dependency paths introduce useful information and that filtering mechanisms are essential. The context representation based on our extended model outperformed other context representations on the context clustering task.

We hypothesized that building the subjective vs. objective distinction into the semantic space will result in more discriminative context representation and thereby in purer context clusters in terms of subjectivity. We tried to modify a semantic space so that it mediates sub-

jectivity. We defined a method where we choose lexicon clues, intensifiers and valenceshifters as dimensions and another one where we choose the dimensions based on their discriminative power. We see that both methods do not improve over the original semantic space.

We define a new algorithm called iterative constrained clustering (ICC) with an active constraint selection strategy. We show that the active selection strategy we propose outperforms previous approach by [Klein et al., 2002], when we utilize them to generate subjectivity sense-tagged data. We also showed that training an SWSD classifier on the semi-automatically acquired data improves over random sampling and uncertainty sampling [Tong and Koller, 2001]. We achieve on MTurkSWSD at least 39% reduction in annotation to train SWSD classifiers of the same accuracy over both sampling strategies. We also represented that the improvements in contextual subjectivity analysis still hold, if we train our SWSD classifiers on semi-automatically generated non-expert labeled data. Overall, the results support our fifth hypothesis:

Hypothesis 5: A "cluster and label" strategy together with some prior knowledge can be utilized to reduce annotation effort to train reliable SWSD classifiers.

5.5 RELATED WORK

Distributional semantic models (DSMs) [Turney and Pantel, 2010, Sahlgren, 2006, Bullinaria and Levy, 2007] have been an important area of research. They have been successfully applied to many NLP tasks. Some examples are word sense discrimination [Schutze, 1998], paraphrase recognition [Lin and Pantel, 2001], thesaurus compilation [Rapp, 2004] and language tests [Landauer and Dutnais, 1997]. DSMs representation word meanings out of context. Recently, several researchers have investigated composition in distributional semantic mod-

els [Erk and Padó, 2008, Reisinger and Mooney, 2010, Mitchell and Lapata, 2010, Rudolph and Giesbrecht, 2010, Grefenstette and Sadrzadeh, 2011]. They demonstrate so far promise paraphrase ranking and phrase similarity rating tasks and offer a powerful tool to represent word meaning in context. Composition is usually achieved through algebraic operations on word vectors or word matrices. Our work relies on the multiplicative model introduced in [Mitchell and Lapata, 2010]. [Mitchell and Lapata, 2010] defines composition between specific word pairs – adjective-noun, noun-noun, verb-object – related over a grammatical dependency relation. We extend their multiplicative model to arbitrary words and grammatical dependencies for general application.

Constrained clustering [Grira et al., 2004] also known as semi-supervised clustering is a recent development in the clustering literature. There are two types of constraints: (1) must-link and (2) cannot-link constraints. There are generally two different strategies to incorporate constraints into the clustering. First strategy is to adapt the underlying distance metric [Xing et al., 2002, Klein et al., 2002] and second strategy is modifying the clustering algorithm itself so that search is biased towards a partitioning for which the constraints hold [Wagstaff and Cardie, 2000, Basu et al., 2002, Demiriz et al., 1999]. Our algorithm is a member of the first strategy and is closely related to [Klein et al., 2002]. [**?**] imposes constraints by changing the distance matrix according to the given constraints. Basically, the distances between must-linked instances are set to 0. We adopt the same strategy to impose hard constraints. In addition, we utilize metric learning to impose soft constraints.

Active selection of constraints for semi-supervised clustering is another related research area. To our knowledge, there have been two previous work selecting constraints for constrained clustering actively [Basu et al., 2004, Klein et al., 2002]. The method described in [Basu et al., 2004] uses the farthest-first traversal scheme for informative selection of pairwise constraints. [Klein et al., 2002] queries constraints between roots of two clusters during

the merging step of hierarchical clustering. In contrast, our active selection strategy queries instance labels and then generate constraints from the labels.

Automatic acquisition of sense-tagged corpora has been investigated in WSD community before. Two main approaches are obtaining training examples via direct Web searching (e.g. [Agirre and Martinez, 2004, Leacock et al., 1998, Mihalcea and Moldovan, 1999, Mihalcea, 2002a]) and via cross-language evidence [Diab, 2004, Chan and Ng, 2005]. To our knowledge, we are the first ones to apply context clustering [Schutze, 1998, Purandare and Pedersen, 2004] for semi-automatic acquisition of coarse-grained sense-tagged data. Web approaches search the Web for instances of a word sense. The search queries are generated based on a dictionary like WordNet. Generally, search queries rely on monosemous – having only one sense – relatives and patterns built from sense definition and synonyms in the dictionary. Although the retrieved examples in this way are high precision, they do not often lead to good supervised WSD performance. [Agirre et al., 2000] explains it with the lack of diversity in the retrieved examples and the distribution of senses in the among the retrieved examples. An notable exception is the work in [Mihalcea, 2002a]. [Mihalcea, 2002a] merges web queries with a bootstrapping approach similar to [Yarowsky, 1995] and achieves performance comparable to manually sense-tagged data, but requires a sense-tagged seed set. Methods on cross-language evidence utilize parallel corpora to obtain sense-tagged examples. For example, [Diab, 2004] groups words which translate to the same target word in a parallel corpora and then map groups to dictionary senses and assign corresponding sense tags to the instances in the corpus. The approaches based on parallel corpora have the disadvantage that parallel corpora are also scarce resources. Our proposed approach is different from previous approaches for automatic acquisition of sense-tagged corpora in the sense that we do not need a dictionary like WordNet. This allows us to handle novel and rare usages of a word as long as they are present in the corpus from which we extract our

examples.

Another related work is on incorporating sentiment content into distributional semantic models [Yessenalina and Cardie, 2011, Maas et al., 2011]. It is a very recent research area. Our work is similar to these approaches in the sense that we try to build external information into the underlying semantic space. Our proposed approach is incorporating subjectivity – subjective vs. objective distinction – while previous work concentrates on polarity – positive vs. negative distinction. Moreover, our proposed approach is unsupervised, while previous work makes use of polarity labelled corpora.

6.0 CONCLUSIONS AND FUTURE DIRECTIONS

This thesis explores methods to utilize sense information to improve contextual subjectivity analysis via sense aware classification. For this purpose, we define a new task *Subjectivity Word Sense Disambiguation* (SWSD) that disambiguates two senses of a word: (1) a subjective sense and (2) an objective sense and feed this information to contextual subjectivity analysis. SWSD aims to capture the right semantic granularity specific to subjectivity analysis. The dissertation is shaped around five main hypotheses.

Hypothesis 1: S/O sense groupings are natural and both groups can be disambiguated accurately by a supervised model.

To test this hypothesis, we introduced the task of subjectivity word sense disambiguation (SWSD), and evaluated a supervised method inspired by research in WSD. The system achieves high accuracy, especially on highly ambiguous words.

Hypothesis 2: The subjectivity sense information provided by SWSD is more reliable than the fine-grained sense information provided by WSD.

To confirm our second hypthesis, we compared the SWSD accuracy to the WSD accuracy on the same dataset. SWSD performs significantly better than WSD. Moreover, SWSD has the advantage to avoid sparsity which will form by utilizing fine-grained senses and also to avoid the dependence on fine-grained sense-tagged data.

Hypothesis 3: SWSD can be exploited to improve the performance of contextual subjectivity analysis systems via sense-aware analysis.

To test our third hypothesis, we applied SWSD in several contextual subjectivity analysis systems, including positive/negative/neutral sentiment classification experimenting with various integration strategies. Significant improvements in performance are realized for all of the target systems.

Hypothesis 4: Crowdsourcing can be utilized to collect high-quality SWSD annotations in order to train SWSD classifiers with a good performance.

To support our fourth hypothesis, we utilized a large pool of non-expert annotators (MTurk) to collect subjectivity sense-tagged data for SWSD. The annotation results support that subjectivity word sense annotation can be done reliably by MTurk workers. In addition, we showed that non-expert annotations are as good as expert annotations for training SWSD classifiers. We demonstrated that SWSD classifiers trained on non-expert annotations improve contextual opinion analysis.

Hypothesis 5: A "cluster and label" strategy together with some prior knowledge can be utilized to reduce annotation effort to train reliable SWSD classifiers.

To confirm our fifth hypothesis, we explored the application of constrained clustering to generate subjectivity sense-tagged data. In order to keep the noise in the semi-automatically labeled data minimal, we experimented with novel techniques to improve cluster purity by improving the context representation and also improving the seed selection strategy for constrained clustering. We achieve 41% reduction in annotation size to train SWSD classifiers of the same accuracy. We also represented that the improvements in contextual subjectivity analysis still hold, if we train our SWSD classifiers on semi-automatically generated non-expert labeled data.

Detailed contributions of this dissertation are summarized below:

- We are the first ones to conceptualize the task Subjectivity Word Sense Disambiguation and use it for sense-aware subjectivity classification. We showed that SWSD is a feasible variant of WSD tailored for our needs.
- We demonstrated that sense-aware analysis enabled by SWSD improves over conventional subjectivity analysis. Our research is a representative of application-specific WSD, which is considered a promising next step in WSD.
- We explored general strategies for SWSD integration. The integration of SWSD information to contextual subjectivity analysis is important. How we do the integration depends on the properties of the underlying system.
- We utilized a large pool of non-expert annotators (MTurk) to collect subjectivity sense-tagged data for SWSD. The annotation results showed that subjectivity word sense annotation can be done reliably by MTurk workers.
- We showed that non-expert annotations are as good as expert annotations for training SWSD classifiers. The additional subjectivity sense-tagged data enabled us to evaluate the benefits of SWSD on contextual subjectivity analysis on a subset of MPQA that is five times larger than senMPQA. We demonstrated that SWSD classifiers trained on non-expert annotations can be exploited to improve contextual opinion analysis.
- We explored the question of whether built-in qualifications are enough to avoid spammers. We had evidence that using more built-in qualification helped to avoid spammers, but our results were not conclusive.
- We investigated the learning effect for workers. Our results showed that there is no improvement in annotator reliability over time for subjectivity sense labeling. For harder annotation tasks (e.g. parse tree annotation) results may be different.

- We explored a "cluster and label" strategy to reduce the human annotation effort needed to generate subjectivity sense-tagged data. We showed that we can achieve a substantial reduction in annotation effort to train SWSD classifiers of the same accuracy over random sampling and uncertainty sampling. The proposed method is not limited to SWSD. We think it is also applicable for the general WSD task.
- We represented that the improvements in contextual subjectivity analysis still hold, if we train our SWSD classifiers on semi-automatically generated non-expert labeled data.
- We extended element-wise multiplication model introduced in [Mitchell and Lapata, 2008] to effectively incorporate richer contexts. Our experiments showed that longer dependency paths introduce useful information and that filtering mechanisms are essential. When we utilize this representation for context clustering, we achieve significant improvement over previous approaches. These results have implications for various lexical disambiguation tasks such as word sense discrimination, paraphrase recognition, and textual entailment.
- We defined a new algorithm called iterative constrained clustering (ICC) for active selection of constraints. We showed that ICC outperforms previous approach by [Klein et al., 2002] on active selection of constraints.

Several questions remain to be answered by future research. Our results are constrained by the coverage of our SWSD system. They imply that a large scale general SWSD component, which can help with various subjectivity and sentiment analysis tasks, is feasible. The natural next step is to obtain training data for remaining subjectivity clues preferably starting with frequent ones. One remaining bottleneck is the generation of subjectivity sense labeled sense inventories, which we show to MTurk workers. Thus, it is worthwhile to evaluate, if MTurk workers can conduct subjectivity sense tagging without the need of subjectivity sense labeled sense inventories. For this purpose, we can give the workers for-

mal online training outside of the MTurk environment and may work with the same set of workers over a long time by creating private groups of workers.

We hypothesized that building the subjective vs. objective distinction into the semantic space will result in more discriminative context representations and thereby in purer context clusters in terms of subjectivity. The methods we tried were not successful. We think there is still hope in this line of research, especially for the second method. Our motivation was to choose the dimensions of a semantic space based on their discriminative power between subjective and objective context of a target word. We utilized automatically generated subjective and objective corpora to provide the signal of the subjective and objective context of the target word. There are a couple of issues with this approach. First of all the automatically generated corpora lack diversity. Second the sentence subjectivity is not a good signal for sense subjectivity for the cases if the word is used with an objective sense. We basically need some annotated instances of the target word in order to disambiguate between its subjective and objective context. It will be a promising step to integrate this dimension selection idea into the ICC. In each iteration, we can select the most discriminative dimensions based on the queried instances we have. This way we will get a more discriminative representation in each iteration.

In our ICC experiment, we fixed the number of cluster to 7 for all target clues. A promising direction will be to learn the number for each target clue separately. There can be a big difference between the numbers of senses two words have. For example, the verb "decide" and the adjective "solid" – two clues in our dataset – have 4 and 14 senses in WordNet, respectively. We think that adjusting cluster number with the usage variety of a word might help to generate more accurate clusters. For this purpose, we can use simple unsupervised cluster evaluation metrics (e.g. silhouette), information criteria (e.g. bayesian information criterion) or non-parametric clustering approaches [Azran and Ghahramani,

2006, Li et al., 2007].

In this research, we strictly work on news documents. It will be interesting to measure the impact of SWSD in other domains or media. We think an important application area is the social networking and micro blogging platforms such as Twitter and Facebook. There is a lot of commercial interest to mine such platforms. These platforms usually pose problems for subjectivity analysis, since the text is usually short and not always grammatical. It will interesting to see if SWSD can help in this context.

BIBLIOGRAPHY

[Agarwal et al., 2009] Agarwal, A., Biadsy, F., and Mckeown, K. (2009). Contextual phrase-level polarity analysis using lexical affect scoring and syntactic N-grams. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 24–32, Athens, Greece. Association for Computational Linguistics.

[Agirre and Edmonds, 2006] Agirre, E. and Edmonds, P., editors (2006). *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*. Springer.

[Agirre and Martinez, 2004] Agirre, E. and Martinez, D. (2004). Unsupervised word sense disambiguation based on automatically retrieved examples: The importance of bias. In *EMNLP 2004*, Barcelona, Spain.

[Agirre et al., 2000] Agirre, E., Rigau, G., Padro, L., and Asterias, J. (2000). Supervised and unsupervised lexical knowledge methods for word sense disambiguation. *Computers and the Humanities*, 34:103–108.

[Akkaya et al., 2010] Akkaya, C., Conrad, A., Wiebe, J., and Mihalcea, R. (2010). Amazon mechanical turk for subjectivity word sense disambiguation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 195–203, Los Angeles. Association for Computational Linguistics.

[Akkaya et al., 2011] Akkaya, C., Wiebe, J., Conrad, A., and Mihalcea, R. (2011). Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 87–96, Portland, Oregon, USA. Association for Computational Linguistics.

[Akkaya et al., 2009] Akkaya, C., Wiebe, J., and Mihalcea, R. (2009). Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in*

Natural Language Processing, pages 190–199, Singapore. Association for Computational Linguistics.

[Akkaya et al., 2012] Akkaya, C., Wiebe, J., and Mihalcea, R. (2012). Utilizing semantic composition in distributional semantic models for word sense discrimination and word sense disambiguation. In *ICSC*, pages 45–51.

[Andreevskaia and Bergler, 2006] Andreevskaia, A. and Bergler, S. (2006). Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings of the 11rd Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*.

[Ar et al., 2011] Ar, B., Joshi, A., and Bhattacharyya, P. (2011). Harnessing wordnet senses for supervised sentiment classification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1081–1091, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[Azran and Ghahramani, 2006] Azran, A. and Ghahramani, Z. (2006). A new approach to data driven clustering. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 57–64, New York, NY, USA. ACM.

[Basu et al., 2002] Basu, S., Banerjee, A., and Mooney, R. (2002). Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning (ICML-2002)*.

[Basu et al., 2004] Basu, S., Banerjee, A., and Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In *SDM*.

[Bullinaria and Levy, 2007] Bullinaria, J. and Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526. 10.3758/BF03193020.

[Callison-Burch, 2009] Callison-Burch, C. (2009). Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295, Morristown, NJ, USA. Association for Computational Linguistics.

[Callison-Burch and Dredze, 2010] Callison-Burch, C. and Dredze, M. (2010). Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12, Los Angeles. Association for Computational Linguistics.

[Chan and Ng, 2005] Chan, Y. S. and Ng, H. T. (2005). Scaling up word sense disambiguation via parallel texts. In *Proceedings of AAAI'05*.

[Davidson et al., 2006] Davidson, I., Wagstaff, K., and Basu, S. (2006). Measuring constraint-set utility for partitional clustering algorithms. In *PKDD*, pages 115–126.

[Davis et al., 2007] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 209–216, New York, NY, USA. ACM.

[Demiriz et al., 1999] Demiriz, A., Bennett, K., and Embrechts, M. J. (1999). Semi-supervised clustering using genetic algorithms. In *In Artificial Neural Networks in Engineering (ANNIE-99*, pages 809–814. ASME Press.

[Diab, 2004] Diab, M. (2004). Relieving the data acquisition bottleneck in word sense disambiguation. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL 2004)*, Barcelona, Spain.

[Dietterich, 1998] Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

[Ding et al., 2008] Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 231–240, New York, NY, USA. ACM.

[Edmonds and Kilgarriff, 2002] Edmonds, P. and Kilgarriff, A. (2002). Introduction to the special issue on evaluating word sense disambiguation systems. *Natural Language Engineering*, 8(4):279–291.

[Erk et al., 2009] Erk, K., McCarthy, D., and Gaylord, N. (2009). Investigations on word senses and word usages. 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*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

[Erk and Padó, 2008] Erk, K. and Padó, S. (2008). A structured vector space model for word meaning in context. In *EMNLP*, pages 897–906.

[Esuli and Sebastiani, 2006a] Esuli, A. and Sebastiani, F. (2006a). Determining term subjectivity and term orientation for opinion mining. In *11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL 2006)*.

[Esuli and Sebastiani, 2006b] Esuli, A. and Sebastiani, F. (2006b). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC-06, the 5th Conference on Language Resources and Evaluation*, Genova, IT.

[Gamon and Aue, 2005] Gamon, M. and Aue, A. (2005). Automatic identification of sentiment vocabulary: Exploiting low association with known sentiment terms. In *Proceedings of the ACL-05 Workshop on Feature Engineering for Machine Learning in Natural Language Processing*, Ann Arbor, US.

[Grefenstette and Sadrzadeh, 2011] Grefenstette, E. and Sadrzadeh, M. (2011). Experimental support for a categorical compositional distributional model of meaning. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

[Grira et al., 2004] Grira, N., Crucianu, M., and Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey. In *in A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence*.

[Gyamfi et al., 2009] Gyamfi, Y., Wiebe, J., Mihalcea, R., and Akkaya, C. (2009). Integrating knowledge for subjectivity sense labeling. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2009)*, pages 10–18, Boulder, Colorado. Association for Computational Linguistics.

[Hatzivassiloglou and McKeown, 1997] Hatzivassiloglou, V. and McKeown, K. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 174–181, Madrid, Spain.

[Hsueh et al., 2009] Hsueh, P.-Y., Melville, P., and Sindhwani, V. (2009). Data quality from crowdsourcing: a study of annotation selection criteria. In *HLT '09: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, Morristown, NJ, USA. Association for Computational Linguistics.

[Hu and Liu, 2004] Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2004 (KDD 2004)*, pages 168–177, Seattle, Washington.

[Jiang et al., 2011] Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting*

of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 151–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Kaisser and Lowe, 2008] Kaisser, M. and Lowe, J. (2008). Creating a research collection of question answer sentence pairs with amazons mechanical turk. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. http://www.lrec-conf.org/proceedings/lrec2008/.

[Kilgarriff, 1997] Kilgarriff, A. (1997). I dont believe in word senses. *Computers and the Humanities*, 31(2):91–113.

[Kilgarriff and Palmer, 2000] Kilgarriff, A. and Palmer, M., editors (2000). *Computer and the Humanities. Special issue: SENSEVAL. Evaluating Word Sense Disambiguation programs*, volume 34.

[Kim and Hovy, 2004] Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the Twentieth International Conference on Computational Linguistics (COLING 2004)*, pages 1267–1373, Geneva, Switzerland.

[Klein et al., 2002] Klein, D., Toutanova, K., Ilhan, I., Kamvar, S., and Manning, C. (2002). Combining heterogeneous classifiers for word-sense disambiguation. In *Proceedings of the ACL Workshop on "Word Sense Disambiguatuion: Recent Successes and Future Directions*, pages 74–80.

[Landauer and Dutnais, 1997] Landauer, T. K. and Dutnais, S. T. (1997). A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240.

[Le et al., 2010] Le, A., Ajot, J., Przybocki, M., and Strassel, S. (2010). Document image collection using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 45–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Leacock and Chodorow, 1998] Leacock, C. and Chodorow, M. (1998). Combining local context and WordNet sense similarity for word sense identification. In *WordNet, An Electronic Lexical Database*. The MIT Press.

[Leacock et al., 1998] Leacock, C., Chodorow, M., and Miller, G. (1998). Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

[Li et al., 2007] Li, J., Ray, S., and Lindsay, B. G. (2007). A nonparametric statistical approach to clustering via mode identification. *J. Mach. Learn. Res.*, 8:1687–1723.

[Lin and Pantel, 2001] Lin, D. and Pantel, P. (2001). Discovery of inference rules for question answering. *Journal of Natural Language Engineering*, 7(3).

[Maas et al., 2011] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

[Martín-Wanton et al., 2010] Martín-Wanton, T., Pons-Porrata, A., Montoyo-Guijarro, A., and Balahur, A. (2010). Opinion polarity detection - using word sense disambiguation to determine the polarity of opinions. In *ICAART 2010 - Proceedings of the International Conference on Agents and Artificial Intelligence, Volume 1*, pages 483–486.

[Mihalcea, 2002a] Mihalcea, R. (2002a). Bootstrapping large sense tagged corpora. In *Proceedings of the Third International Conference on Language Resources and Evaluation LREC 2002*, pages 1407–1411, Canary Islands, Spain.

[Mihalcea, 2002b] Mihalcea, R. (2002b). Instance based learning with automatic feature selection applied to Word Sense Disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

[Mihalcea and Edmonds, 2004] Mihalcea, R. and Edmonds, P., editors (2004). *Proceedings of SENSEVAL-3, Association for Computational Linguistics Workshop*, Barcelona, Spain.

[Mihalcea and Moldovan, 1999] Mihalcea, R. and Moldovan, D. (1999). An automatic method for generating sense tagged corpora. In *Proceedings of AAAI-99*, pages 461–466, Orlando, FL.

[Mihalcea and Moldovan, 2001] Mihalcea, R. and Moldovan, D. (2001). EZ.WordNet: principles for automatic generation of a coarse grained WordNet. In *Proceedings of FLAIRS-2001*, pages 454–458, Key West.

[Miller, 1995] Miller, G. (1995). Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.

[Mitchell and Lapata, 2008] Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

[Mitchell and Lapata, 2010] Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

[Mrozinski et al., 2008] Mrozinski, J., Whittaker, E., and Furui, S. (2008). Collecting a why-question corpus for development and evaluation of an automatic QA-system. In *Proceedings of ACL-08: HLT*, pages 443–451, Columbus, Ohio. Association for Computational Linguistics.

[Navigli, 2006] Navigli, R. (2006). Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

[Negri et al., 2011] Negri, M., Bentivogli, L., Mehdad, Y., Giampiccolo, D., and Marchetti, A. (2011). Divide and conquer: Crowdsourcing the creation of cross-lingual textual entailment corpora. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 670–679, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[Palmer et al., 2004] Palmer, M., Babko-Malaya, O., and Dang, H. T. (2004). Different sense granularities for different applications. In *HLT-NAACL 2004 Workshop: 2nd Workshop on Scalable Natural Language Understanding*, Boston, Massachusetts.

[Parent and Eskenazi, 2010] Parent, G. and Eskenazi, M. (2010). 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*, pages 21–29, Los Angeles. Association for Computational Linguistics.

[Passonneau et al., 2006] Passonneau, R., Habash, N., and Rambow, O. (2006). Inter-annotator agreement on a multilingual semantic annotation task. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*.

[Pradhan and Xue, 2009] Pradhan, S. S. and Xue, N. (2009). Ontonotes: The 90% solution. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*, pages 11–12, Boulder, Colorado. Association for Computational Linguistics.

[Preiss and Yarowsky, 2001] Preiss, J. and Yarowsky, D., editors (2001). *Proceedings of SENSEVAL-2, Association for Computational Linguistics Workshop*, Toulouse, France.

[Purandare and Pedersen, 2004] Purandare, A. and Pedersen, T. (2004). Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2004)*, Boston.

[Quirk et al., 1985] Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language.* Longman, New York.

[Rapp, 2004] Rapp, R. (2004). *A freely available automatically generated thesaurus of related words*, pages 395–398.

[Reisinger and Mooney, 2010] Reisinger, J. and Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *HLT-NAACL*, pages 109–117.

[Rentoumi et al., 2009] Rentoumi, V., Giannakopoulos, G., Karkaletsis, V., and Vouros, G. A. (2009). Sentiment analysis of figurative language using a word sense disambiguation approach. In *Proceedings of the International Conference RANLP-2009*, pages 370–375, Borovets, Bulgaria. Association for Computational Linguistics.

[Riloff and Wiebe, 2003] Riloff, E. and Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, pages 105–112, Sapporo, Japan.

[Riloff et al., 2005] Riloff, E., Wiebe, J., and Phillips, W. (2005). Exploiting subjectivity classification to improve information extraction. In *Proc. 20th National Conference on Artificial Intelligence (AAAI-2005)*, pages 1106–1111, Pittsburgh, PA.

[Rosner, 2006] Rosner, B. (2006). *Fundamentals of Biostatistics.* Thompson Brooks/Cole.

[Rudolph and Giesbrecht, 2010] Rudolph, S. and Giesbrecht, E. (2010). Compositional matrix-space models of language. In *ACL*, pages 907–916.

[Sahlgren, 2006] Sahlgren, M. (2006). *The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* PhD thesis, Stockholm University.

[Schapire and Singer, 2000] Schapire, R. E. and Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

[Schutze, 1998] Schutze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

[Schütze et al., 2006] Schütze, H., Velipasaoglu, E., and Pedersen, J. O. (2006). Performance thresholding in practical text classification. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 662–671, New York, NY, USA. ACM.

[Snow et al., 2008] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Morristown, NJ, USA. Association for Computational Linguistics.

[Snow et al., 2007] Snow, R., Prakash, S., Jurafsky, D., and Ng, A. (2007). Learning to merge word senses. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic.

[Sorokin and Forsyth, 2008] Sorokin, A. and Forsyth, D. (2008). Utility data annotation with amazon mechanical turk. pages 1 –8.

[Stoyanov et al., 2005] Stoyanov, V., Cardie, C., and Wiebe, J. (2005). Multi-Perspective Question Answering using the OpQA corpus. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 923–930, Vancouver, Canada.

[Su and Markert, 2008] Su, F. and Markert, K. (2008). From word to sense: a case study of subjectivity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-2008)*, Manchester.

[Su and Markert, 2009] Su, F. and Markert, K. (2009). Subjectivity recognition on word senses via semi-supervised mincuts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1–9, Boulder, Colorado. Association for Computational Linguistics.

[Su and Markert, 2010] Su, F. and Markert, K. (2010). Word sense subjectivity for cross-lingual lexical substitution. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 357–360, Los Angeles, California. Association for Computational Linguistics.

[Tan et al., 2005] Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Thater et al., 2009] Thater, S., Dinu, G., and Pinkal, M. (2009). Ranking paraphrases in context. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, TextInfer '09, pages 44–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Tong and Koller, 2001] Tong, S. and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66.

[Turney, 2002a] Turney, P. (2002a). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 417–424, Philadelphia, Pennsylvania.

[Turney, 2002b] Turney, P. (2002b). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *(ACL 2002)*, pages 417–424, Philadelphia.

[Turney and Pantel, 2010] Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics.

[Wagstaff and Cardie, 2000] Wagstaff, K. and Cardie, C. (2000). Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, pages 1103–1110.

[Whitelaw et al., 2005] Whitelaw, C., Argamon, S., and Garg, N. (2005). Using appraisal taxonomies for sentiment analysis. In *Proceedings of the First Computational Systemic Functional Grammar Conference.*

[Wiebe, 1994] Wiebe, J. (1994). Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

[Wiebe, 2002] Wiebe, J. (2002). Instructions for annotating opinions in newspaper articles. Department of Computer Science Technical Report TR-02-101, University of Pittsburgh.

[Wiebe et al., 1999] Wiebe, J., Bruce, R., and O'Hara, T. (1999). Development and use of a gold standard data set for subjectivity classifications ann. In *Proceedings of the 37th*

Annual Meeting of the Association for Computational Linguistics (ACL-99), pages 246–253, College Park, Maryland.

[Wiebe and Mihalcea, 2006] Wiebe, J. and Mihalcea, R. (2006). Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1065–1072, Sydney, Australia. Association for Computational Linguistics.

[Wiebe and Riloff, 2005] Wiebe, J. and Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005) (invited paper)*, Mexico City, Mexico.

[Wiebe et al., 2004] Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30(3):277–308.

[Wiebe et al., 2005a] Wiebe, J., Wilson, T., and Cardie, C. (2005a). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

[Wiebe et al., 2005b] Wiebe, J., Wilson, T., and Cardie, C. (2005b). Annotating expressions of opinions and emotions in language ann. *Language Resources and Evaluation*, 39(2/3):164–210.

[Wilson, 2007] Wilson, T. (2007). *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of private states*. PhD thesis, Intelligent Systems Program, University of Pittsburgh.

[Wilson et al., 2005] Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 347–354, Vancouver, Canada.

[Witten and Frank., 2005] Witten, I. and Frank., E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann.

[Xing et al., 2002] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. J. (2002). Distance metric learning with application to clustering with side-information. In *NIPS*, pages 505–512.

[Yano et al., 2010] Yano, T., Resnik, P., and Smith, N. A. (2010). 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*, pages 152–158, Los Angeles. Association for Computational Linguistics.

[Yarowsky, 1995] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995)*, Cambridge, MA.

[Yessenalina and Cardie, 2011] Yessenalina, A. and Cardie, C. (2011). Compositional matrix-space models for sentiment analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 172–182, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[Yu and Hatzivassiloglou, 2003] Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, pages 129–136, Sapporo, Japan.

[Zhai et al., 2011] Zhai, Z., Liu, B., Zhang, L., Xu, H., and Jia, P. (2011). Identifying evaluative sentences in online discussions. In Burgard, W. and Roth, D., editors, *AAAI*. AAAI Press.