# Modeling and Representing Negation in Data-driven Machine Learning-based Sentiment Analysis

Robert Remus

Natural Language Processing Group,
Department of Computer Science,
University of Leipzig, Germany
rremus@informatik.uni-leipzig.de

**Abstract.** We propose a scheme for explicitly modeling and representing negation of word $n$-grams in an augmented word $n$-gram feature space. For the purpose of negation scope detection, we compare 2 methods: the simpler regular expression-based NegEx, and the more sophisticated Conditional Random Field-based LingScope. Additionally, we capture negation implicitly via word bi- and trigrams. We analyze the impact of explicit and implicit negation modeling as well as their combination on several data-driven machine learning-based sentiment analysis subtasks, i.e. document-level polarity classification, both in- and cross-domain, and sentence-level polarity classification. In all subtasks, explicitly modeling negation yields statistically significant better results than not modeling negation or modeling it only implicitly.

**Keywords:** Sentiment analysis, negation modeling, machine learning

## 1 Introduction

Negations as in example (1)

(1) *Don't* ask me!

are at the core of human language. Hence, negations are commonly encountered in natural language processing (NLP) tasks, e.g. textual entailment [1, 2]. In sentiment analysis (SA), negation plays a special role [3]: Whereas example (2) expresses positive sentiment, the only slightly different example (3) expresses negative sentiment.

(2) They are $\langle$comfortable to wear$\rangle^+$.

(3) They are $\langle$*not* $\langle$~~comfortable to wear~~$\rangle^+\rangle^-$.[1]

[1] In this work, struck out ~~words~~ are considered as negated.

Therefore, negations are frequently treated in compositional semantic approaches to SA [4–8], as well as in bag of words-based machine learning (ML) techniques [9, 10].

Research on negation scopes (NSs) and negation scope detection (NSD) was primarily driven by biomedical NLP, particularly research on the detection of absence or presence of certain diseases in biomedical text. One of the most prominent studies in this field is [11], that identifies negation words and their scope using a variety of ML techniques and features. Only quite recently, the impact of NSD on SA became of increasing interest: [12–14] detect NSs using parse trees, typed dependencies, semantic role labeling and/or manually defined negation words. [15] compare several baselines for NSD, e.g. they consider as NS the rest of the sentence following a negation word, or a fixed window of 1 to 4 words following, preceding or around a negation word. [16, 17] study NSD based on Conditional Random Fields (CRFs). All these studies concur in their conclusion that SA, or more precisely *polarity classification*, benefits from NSD.

We model NSs in word $n$-gram feature space systematically and adopt recent advances in NSD. We believe this endeavor is worthwhile, as this allows machines to learn by themselves how negations modify the *meaning* of words, instead of being taught by manually defined and often ad hoc rules. Our work focuses on data-driven ML-based models for SA that operate in word $n$-gram feature space and do not rely on lexical resources, e.g. prior polarity dictionaries like *SentiWordNet* [18]. While various methods and features have been proposed for SA, such data-driven word $n$-gram models proved to be still competitive in many recent studies [19–21].

This paper is structured as follows: In the next section we describe our approach to modeling and representing negation in data-driven ML-based SA. In Sect. 3 we evaluate our approach in experiments for several SA subtasks and discuss their results. Finally, we draw conclusions and point out possible directions for future work in Sect. 4.

## 2 Negation Modeling

We now describe our approach to implicitly and explicitly modeling and representing negation in word $n$-gram feature space for data-driven ML-based SA. When *explicitly* modeling negation, we incorporate our knowledge of negation into the model; when *implicitly* modeling negation, we do not.

### 2.1 Implicit Negation Modeling

As pointed out in [3], negations are often *implicitly* modeled via higher order word $n$-grams, e.g. bigrams ("*n't* return"), trigrams ("*lack of* padding"), tetragrams[2] ("*denied* sending wrong size") etc. That aside, higher order word $n$-grams also implicitly capture other linguistic phenomena, e.g. comparatives ("larger than", "too much").

[2] Tetragrams are also referred to as quad-, four- or 4-grams.

### 2.2 Explicit Negation Modeling

Although it is convenient, there is a drawback to solely relying on higher order word $n$-grams when trying to capture negations: Long NSs as in example (4) occur frequently (cf. Sect. 3.3), but typically word $n$-grams ($n < 5$) are not able to properly capture them.

(4) The leather straps have *never* ~~worn out or broken~~.

Here, a word trigram captures "never worn out" but not "never [..] broken". While a word 5-gram is able to capture "never [..] broken", learning models using word $n$-gram features with $n \geq 3$ usually leads to very sparse representations, depending on how much training data is available and how homogeneous [22] this training data is. In such cases, learning from the training data what a certain higher order word $n$-gram contributes to the model is then backed up by only very little to almost none empirical findings. Therefore, we model negations also *explicitly*.

**Negation Scope Detection** Vital to explicit negation modeling is NSD. E.g., in example (5), we need to detect that "stand up to laundering very well" is in the scope of "don't".

(5) They *don't* ~~stand up to laundering very well~~, in that they shrink up quite a bit.

For that purpose, we employ NegEx[3] [23], a simpler regular expression-based NSD and LingScope[4] [24], a more sophisticated CRF-based NSD trained on the BioScope corpus [25]. NegEx was chosen as a strong baseline: its detected NSs are similar to a weak baseline NSD method frequently used [9, 10]: consider all words following a negation word as negated, up to the next punctuation. LingScope was chosen to represent the state-of-the-art in NSD. Additionally, both NegEx and LingScope are publicly available.

To improve NSD, we expand contractions like "can't" to "can not", "didn't" to "did not" etc. Please note that while NegEx considers the negation itself to be part of the NS, we do not. NegEx's NSs are adjusted accordingly.

**Representation in Feature Space** Once NSs are detected, negated and non-negated word $n$-grams need to be explicitly represented in feature space. Therefore, we resort to a representation inspired by [9], who create a new feature `NOT_f` when feature `f` is preceded by a negation word, e.g. "not" or "isn't".

Let $\mathcal{W} = \{w_i\}, i = 1, \ldots, d$ be our word $n$-grams and let $\mathcal{X} = \{0, 1\}^d$ be our word $n$-gram feature space of size $d$, where for $x_j \in \mathcal{X}$, $x_{j_k} = 1$ denotes the presence of $w_k$ and $x_{j_k} = 0$ denotes its absence. For each feature $x_{j_k}$ we introduce an additional feature $\breve{x}_{j_k}$ that encodes whether $w_k$ appears negated ($\breve{x}_{j_k} = 1$) or non-negated ($\breve{x}_{j_k} = 0$). Thus, we obtain an augmented feature space $\breve{\mathcal{X}} = \{0, 1\}^{2d}$. In $\breve{\mathcal{X}}$ we are now able to represent whether a word $n$-gram

[3] `http://code.google.com/p/negex/`

[4] `http://sourceforge.net/projects/lingscope/`

**Table 1.** Representation of example (5) in $\breve{\mathcal{X}}$ as described in Sect. 5.

| bit | don't | down | ~~laundering~~ | quite | shrink | ~~stand~~ | ~~up~~/up | ~~very~~ | ~~well~~ |
|---|---|---|---|---|---|---|---|---|---|
| $[1,0$ | $1,0$ | $0,0$ | $0,1$ | $1,0$ | $1,0$ | $0,1$ | $1,1$ | $0,1$ | $0,1]$ |

- $w$ is present (encoded as $[1,0]$),
- $w$ is absent ($[0,0]$),
- $w$ is present and negated ($[0,1]$) or
- $w$ is present both negated and non-negated ($[1,1]$).

**Representing an Example** Assume we employ naïve tokenization that simply splits at white spaces, ignore punctuation characters like "." and "," and extract the presence and absence of the word unigrams $\mathcal{W}_{uni} = \{$"bit", "don't", "down", "laundering", "quite", "shrink", "stand", "up", "very", "well"$\}$, i.e. $\mathcal{W}_{uni}$ is our *vocabulary*. Representing example (5) in $\breve{\mathcal{X}}$ results then in a stylized feature vector as shown in Table 1.

Note the difference between "laundering" and "up". While "laundering" is present only once and is negated and thus is represented as $[0,1]$, "up" is present twice—once negated and once non-negated—and thus is represented as $[1,1]$.

## 3 Evaluation

We evaluate our negation modeling approach in 3 common SA subtasks: in-domain document-level polarity classification, cross-domain document-level polarity classification (cf. Sect. 3.1) and sentence-level polarity classification (cf. Sect. 3.2).

Our setup for all experiments is as follows: For sentence segmentation and tokenization we use OpenNLP[5]. As classifiers we employ Support Vector Machines (SVMs) in their LibSVM implementation[6] using a linear kernel with their cost factor $C$ set to 2.0 without any further optimization. SVMs were chosen because (i) it has been shown previously that they exhibit superior classification power in polarity classification experiments [9] and therefore (ii) nowadays SVMs are a common choice for SA classification subtasks and text classification in general [26].

As features we use word uni-, bi- and trigrams extracted from the data[7]. Word bi- and trigrams model negation implicitly as described in Sect. 2.1. We

[5] `http://opennlp.apache.org`

[6] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[7] We also experimented with word tetragrams, but found that they do not contribute to the models' discriminative power. This is not surprising, as in all used data sets most word tetragrams appear only once. The word tetragram distribution's *relative entropy* [27], is greater than 0.99, i.e. here word tetragrams are almost uniformly distributed.

perform no feature selection—neither stop words nor punctuation characters are removed because we do not make any assumption about which word $n$-grams carry sentiment and which do not. Additionally, we explicitly model the negation of these word uni-, bi- and trigrams as described in Sect. 2.2. This is different from [9]'s approach, who "[..] consider bigrams (and $n$-grams in general) to be an orthogonal way to incorporate context.". Explicitly modeling negation of higher order word $n$-grams allows for learning that there is a difference between "doesn't work well" and "doesn't work" in examples (6) and (7)

(6) The stand *doesn't* ~~work~~ ~~well~~.

(7) The stand *doesn't* ~~work~~.

just as an ordinary word {uni, bi}-gram model allows for learning the difference between "work" and "work well".

The in-domain document-level and sentence-level polarity classification experiments are construed as 10-fold cross validations. As performance measure we report accuracy $A$ to be comparable to other studies (cf. Sect. 3.4). The level of statistical significance is determined by *stratified shuffling*, an approximate randomization test [28] run with $2^{20} = 1,048,576$ iterations as recommended by [29]. The level of statistically significant difference to the corresponding base model without negation modeling is indicated by $\star\star$ ($p < 0.005$) and $\star$ ($p < 0.05$).

### 3.1 Document-level Polarity Classification

As gold standard for in- and cross-domain document-level polarity classification we use [30]'s Multi-domain Sentiment Dataset v2.0[8] (MDSD v2.0), that contains star-rated product *reviews* of various domains. We chose 10 domains: apparel, books, dvd, electronics, health & personal care, kitchen & housewares, music, sports & outdoors, toys & games and video. Those are exactly the domains for which a pre-selected, balanced amount of 1,000 positive and 1,000 negative reviews is available. [30] consider reviews with more than 3 stars positive, and less than 3 stars negative—they omit 3-star reviews; so do we.

**In-domain** The evaluation results of our in-domain document-level polarity classification experiments averaged over all 10 domains are shown in Table 2.

A word {uni, bi}-gram base model, LingScope for NSD and explicitly modeling negations for word {uni, bi}-grams yields the best overall result ($A = 81.93$). This result is statistically significant different ($p < 0.005$) from the result the corresponding base model achieves using word {uni, bi}-grams alone ($A = 81.37$).

**Cross-domain** In our cross-domain experiments, for all $^{10!}/_{(10\text{-}2)!} = 90$ source domain–target domain pairs, there are 2,000 labeled source domain instances (1,000 positive and 1,000 negative) and 200 labeled target domain instances (100

[8] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

**Table 2.** Accuracies for in-domain document-level polarity classifications, averaged over 10 domains from MDSD v2.0.

| Base model | NSD method | Explicit negation modeling for | | |
|---|---|---|---|---|
| | | {uni} | {uni, bi} | {uni, bi, tri} |
| {uni} | none | 78.77 | | |
| | LingScope | 80.06 ⋆⋆ | | |
| | NegEx | 79.57⋆ | | |
| {uni, bi} | none | 81.37 | | |
| | LingScope | 81.73 | **81.93** ⋆⋆ | |
| | NegEx | 81.53 | 81.58 | |
| {uni, bi, tri} | none | 81.27 | | |
| | LingScope | 81.65⋆ | 81.55 | 81.59⋆ |
| | NegEx | 81.28 | 81.3 | 81.28 |

positive and 100 negative) available for training, 1,800 labeled target domain instances (900 positive and 900 negative) are used for testing. This is a typical *semi-supervised domain adaptation* setting. If required by the method the same amount of unlabeled target domain instances is available for training as there are labeled source domain instances: 2,000.

We employ 3 methods for cross-domain polarity classification, Instance Selection (IS) [31], “All” and EasyAdapt++ (EA++) [32]. While “All” simply uses all available labeled source and target domain training instances for training, EA++ additionally uses unlabeled target domain instances and operates via feature space augmentation and co-regularization [33]. IS selects source domain training instances that are most likely to be informative based on domain similarity and domain complexity of source and target domain.

Table 3 shows the evaluation results for “All”. Due to space restrictions, we only present the best results for IS and EA++. Full evaluation results are available at the authors’ website[9].

For “All”, just like for in-domain polarity classification, a word {uni, bi}-gram base model, LingScope for NSD and explicitly modeling negations for word {uni, bi}-grams yields the best overall result ($A = 77.31$, $p < 0.005$). The same applies to IS ($A = 77.71$, $p < 0.005$). For EA++, a word {uni, bi}-gram base model, NegEx for NSD and explicitly modeling negations for word unigrams yields the best overall result ($A = 77.5$, $p < 0.005$). A word {uni, bi, tri}-gram base model, LingScope for NSD and explicitly modeling negations for word unigrams performs almost as good and yields $A = 77.48$ ($p < 0.005$).

[9] `http://asv.informatik.uni-leipzig.de/staff/Robert_Remus`

**Table 3.** Accuracies for cross-domain document-level polarity classification ("All"), averaged over 90 domain-pairs from MDSD v2.0.

| **Base model** | **NSD method** | **Explicit negation modeling for** | | |
|---|---|---|---|---|
| | | {uni} | {uni, bi} | {uni, bi, tri} |
| {uni} | none | 74.25 | | |
| | LingScope | 75.46 ⋆⋆ | | |
| | NegEx | 75.35 ⋆⋆ | | |
| {uni, bi} | none | 76.61 | | |
| | LingScope | 77.23 ⋆⋆ | **77.31** ⋆⋆ | |
| | NegEx | 77.18 ⋆⋆ | 77.13 ⋆⋆ | |
| {uni, bi, tri} | none | 76.44 | | |
| | LingScope | 77.01 ⋆⋆ | 77.13 ⋆⋆ | 77.12 ⋆⋆ |
| | NegEx | 76.97 ⋆⋆ | 76.83 ⋆⋆ | 76.81 ⋆⋆ |

**Table 4.** Accuracies for sentence-level polarity classification of SPD v1.0.

| **Base model** | **NSD method** | **Explicit negation modeling for** | | |
|---|---|---|---|---|
| | | {uni} | {uni, bi} | {uni, bi, tri} |
| {uni} | none | 74.56 | | |
| | LingScope | 75.85 ⋆⋆ | | |
| | NegEx | 75.08 | | |
| {uni, bi} | none | 77.69 | | |
| | LingScope | 77.93 | 77.55 | |
| | NegEx | 77.72 | 77.36 | |
| {uni, bi, tri} | none | 77.62 | | |
| | LingScope | 77.85 | 77.99 | **78.01**⋆ |
| | NegEx | 77.71 | 77.23 | 77.36 |

### 3.2 Sentence-level Polarity Classification

As gold standard for sentence-level polarity classification we use [34]'s sentence polarity dataset v1.0[10] (SPD v1.0), that contains 10,662 sentences from movie *reviews* annotated for their polarity (5,331 positive and 5,331 negative).

Evaluation results are shown in Table 4. Here, a word {uni, bi, tri}-gram base model, LingScope for NSD and explicitly modeling negations for word {uni, bi, tri}-grams yields the best result ($A = 78.01$, $p < 0.05$).

### 3.3 Discussion

Intuitively, explicit negation modeling benefits from high quality NSD: The more accurate the NSD, the more accurate the explicit negation modeling. This intuition is met by our results. As shown by [24], LingScope is often more accurate

[10] `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

**Table 5.** Evaluation results of LingScope and NegEx on SPD v1.0.

| **NSD method** | **Precision** | **Recall** | **F-Score** |
|---|---|---|---|
| LingScope | 0.696 | 0.656 | 0.675 |
| NegEx | 0.407 | 0.5 | 0.449 |

**Table 6.** Negation scope statistics. # number of NSs, $\bar{\#}$ average number of NSs per document/sentence, $w/$ percentage of documents/sentences with detected NSs, $\bar{l}$ average NS length in tokens, $l = 1, 2, 3, \geq 4$ distribution of NSs of the according length.

| **Data set** | **NSD** | # | $\bar{\#}$ | $w/$ | $\bar{l}$ | $l = 1$ | $l = 2$ | $l = 3$ | $l \geq 4$ |
|---|---|---|---|---|---|---|---|---|---|
| MDSD v2.0 | LingScope | 3,187.5 | 1.6 | 67.4% | 6.6 | 1.4% | 13.5% | 12.7% | 72.5% |
|  | NegEx | 2,971.2 | 1.5 | 67.3% | 10.7 | 1.8% | 6.6% | 8.4% | 83.2% |
| SPD v1.0 | LingScope | 2,339 | 0.2 | 20.5% | 6.8 | 2.2% | 9.8% | 13.8% | 74.2% |
|  | NegEx | 2,085 | 0.2 | 19.6% | 12.1 | 1.9% | 3.8% | 5.9% | 88.3% |

than NegEx on *biomedical* data. This also applies to *review* data: We evaluated LingScope and NegEx on 500 sentences that were randomly extracted from SPD v1.0 and annotated for their NSs. Table 5 shows the results: LingScope clearly outperforms NegEx with respect to precision and recall. So although BioScope's genre domain which LingScope and NegEx were trained and/or tested on differs greatly from the genre and domains of MDSD v2.0 and SPD v1.0, models learned using LingScope yield the best or almost best results for all our SA subtasks.

Compared to ordinary word $n$-gram models that do not model negation ($n = 1$) or model negation only implicitly ($2 \leq n \leq 3$), word $n$-gram models that additionally model negation explicitly achieve statistically significant improvements—given an accurate NSD method.

To shed some light on the differences between the evaluated subtasks' and gold standards' results, we analyze how many and what kind of NSs the NSD methods detect (cf. Table 6). Generally, LingScope detects more negations than NegEx. NSs detected by LingScope are on average shorter than those detected by NegEx, hence they are more precise. While LingScope and NegEx detect negations in about 67% of all documents in MDSD v2.0, only about 20% of all sentences in SPD v1.0 contain detected negations.

It is noteworthy that only very little NSs have length 1, i.e. span 1 word unigram, but many NSs have length 4 or longer, i.e. span 4 word unigrams or more. That confirms the need for explicit negation modeling as mentioned in Sect. 2.2, but also hints at a data sparsity problem: Parts of word $n$-grams in the scope of negations re-occur, but the same NS basically never appears twice. E.g., for MDSD v2.0 and LingScope as NSD, on average each NS overlaps only on 0.18 positions with each other NS. Thus, overlaps as shown in example (8) and (9), where "buy" appears in both NSs, are scarce:

(8) *Don't* ~~buy these shoes for running~~!

(9) Do *not* ~~buy~~ ~~them~~ unless you like getting blisters.

The picture is similar for SPD v1.0 with an overlap in 0.22 positions on average.

### 3.4 Comparison

For sentence-level polarity classification on SPD v1.0 our best performing model ($A = 78.01$) outperforms 3 state-of-the-art models: [35]'s dependency tree-based CRFs with hidden variables ($A = 77.3$), [8]'s linear Matrix-Vector Recursion ($A = 77.1$) and [36] Semi-supervised Recursive Autoencoders ($A = 77.7$). It is only beaten by [8]'s matrix-vector recursive neural network ($A = 79$) and [37]'s SVM with naïve bayes features ($A = 79.4$).

For in-domain document-level polarity classification on MDSD v2.0, [27] report results for 7 domains (dvd, books, electronics, health, kitchen, music, toys) out of the 10 domains we used in our experiments. Their SVMs use word unigrams and bigrams of word stems as features and yield $A = 80.29$ on average; on the same 7 domains our best performing model yields $A = 81.49$ on average.

For cross-domain document-level polarity classification on MDSD v2.0, our best performing model (IS, $A = 76.76$) is inferior compared to more complex domain adaptation methods, all of which are *evaluated on 4 domains* (dvd, books, electronics, kitchen), i.e. 12 domain pairs: [30]'s Structural Correspondence Learning ($A = 77.97$), [38]'s Spectral Feature Alignment ($A = 78.75$) and [39]'s graph-based RANK ($A = 76.9$), OPTIM-SOCAL ($A = 76.78$) and RANK-SOCAL ($A = 80.12$). It only outperforms [39]'s OPTIM ($A = 75.3$).

In summary, purely data-driven discriminative word $n$-gram models with negation modeling prove to be competitive in several common SA subtasks.

## 4 Conclusions & Future Work

We conclude that data-driven ML-based models for SA that operate in word $n$-gram feature space benefit from explicit negation modeling. In turn, explicit negation modeling benefits from (i) high quality NSD methods like LingScope and (ii) modeling not only negation of word unigrams, but also of higher order word $n$-grams, especially word bigrams.

These insights suggest that explicitly modeling semantic compositions is promising for data-driven ML-based SA. Given appropriate scope detection methods, our approach may for example easily be extended to model other *valence shifters* [40], e.g. intensifiers like "very" or "many", or *hedges* [41] like "may" or "might", or even implicit negation in the absence of negation words [42]. Our approach is also easily extensible to other word $n$-gram weighting schemes aside from encoding pure presence or absence, e.g. weighting using relative frequencies or tf-idf. The feature space then simply becomes $\check{\mathcal{X}} = \mathbb{R}^{2d}$.

Future work encompasses model fine-tuning, e.g. accounting for NSs in the scope of other negations as in example (10)

(10) I ⟨*don't* ~~care~~ ~~that~~ ~~they~~ ~~are~~ ⟨~~*not*~~ ~~really~~ ~~leather~~⟩⟩.

and employing generalization methods to tackle data sparsity when learning the effects of negations, modeled both implicitly and explicitly.

## Acknowledgments

A special thank you goes to Stefan Bordag for the fruitful discussions we had. Additional thanks goes to the anonymous reviewers whose useful comments and suggestions considerably improved the original paper.

## References

1. Herrera, J., Penas, A., Verdejo, F.: Textual entailment recognition based on dependency analysis and wordnet. In: Proceedings of the 1st PASCAL Challenge Workshop on Recognizing Textual Entailment (RTE). (2005) 21–24
2. Delmonte, R., Tonelli, S., Boniforti, M.A.P., Bristot, A.: Venses – a linguistically-based system for semantic evaluation. In: Proceedings of the 1st PASCAL Challenge Workshop on Recognizing Textual Entailment (RTE). (2005) 49–52
3. Wiegand, M., Balahur, A., Roth, B., Klakow, D., Montoyo, A.: A survey on the role of negation in sentiment analysis. In: Proceedings of the 2010 Workshop on Negation and Speculation in Natural Language Processing (NeSp-NLP). (2010) 60–68
4. Moilanen, K., Pulman, S.: Sentiment composition. In: Proceedings of the 6th International Conference on Recent Advances in Natural Language Processing (RANLP). (2007) 378–382
5. Choi, Y., Cardie, C.: Learning with compositional semantics as structural inference for subsentential sentiment analysis. In: Proceedings of the 13th Conference on Empirical Methods in Natural Language Processing (EMNLP). (2008) 793–801
6. Neviarouskaya, A., Prendinger, H., Ishizuka, M.: Compositionality principle in recognition of fine-grained emotions from text. In: Proceedings of the 3rd International Conference on Weblogs and Social Media (ICWSM). (2009) 278–281
7. Remus, R., Hänig, C.: Towards well-grounded phrase-level polarity analysis. In: Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing). (2011) 380–392
8. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL). (2012) 1201–1211
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL). (2002) 79–86
10. Mohammad, S., Kiritchenko, S., Zhu, X.: Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval). (2013) 321–327
11. Morante, R., Daelemans, W.: A metalearning approach to processing the scope of negation. In: Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL). (2009) 21–29

12. Jia, L., Yu, C., Meng, W.: The effect of negation on sentiment analysis and retrieval effectiveness. In: Proceedings of the 18th Conference on Information and Knowledge Management (CIKM). (2009) 1827–1830
13. Carrillo-de Albornoz, J., Plaza, L.: An emotion-based model of negation, intensifiers, and modality for polarity and intensity classification. Journal of the American Society for Information Science and Technology (JASIST) (2013)
14. Johansson, R., Moschitti, A.: Relational features in fine-grained opinion analysis. Computational Linguistics **39**(3) (2013)
15. Hogenboom, A., van Iterson, P., Heerschop, B., Frasincar, F., Kaymak, U.: Determining negation scope and strength in sentiment analysis. In: Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC). (2011) 2589–2594
16. Councill, I.G., McDonald, R., Velikovich, L.: What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In: Proceedings of the 2010 Workshop on Negation and Speculation in Natural Language Processing (NeSp-NLP). (2010) 51–59
17. Lapponi, E., Read, J., vrelid, L.: Representing and resolving negation for sentiment analysis. In: Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012) Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction (SENTIRE). (2012) 687–692
18. Esuli, A., Sebastiani, F.: SentiWordNet: A publicly available lexical resource for opinion mining. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC). (2006) 417–422
19. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING). (2010) 36–44
20. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media (LSM. (2011) 30–38
21. Saif, H., He, Y., Alani, H.: Alleviating data sparsity for twitter sentiment analysis. In: Proceedings of the 2nd Workshop on Making Sense of Microposts (#MSM). (2012)
22. Kilgarriff, A.: Comparing corpora. International Journal of Corpus Linguistics **6**(1) (2001) 97–133
23. Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F., Buchanan, B.G.: A simple algorithm for identifying negated findings and diseases in discharge summaries. Journal of Biomedical Informatics **34**(5) (2001) 301–310
24. Agarwal, S., Yu, H.: Biomedical negation scope detection with conditional random fields. Journal of the American Medical Informatics Association **17**(6) (2010) 696–701
25. Vincze, V., Szarvas, G., Farkas, R., Móra, G., Csirik, J.: The BioScope corpus: Biomedical texts annotated for uncertainty, negation and their scopes. BMC Bioinformatics **9**(Suppl 11) (2008) S9
26. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning (ECML). (1998) 137–142
27. Ponomareva, N., Thelwall, M.: Biographies or blenders: Which resource is best for cross-domain sentiment analysis? In: Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing). (2012) 488–499

28. Noreen, E.W.: Computer Intensive Methods for Testing Hypothesis – An Introduction. John Wiley and Sons, Inc. (1989)
29. Yeh, A.: More accurate tests for the statistical significance of result differences. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING). (2000) 947–953
30. Blitzer, J., Dredze, M., Pereira, F.C.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL). (2007) 440–447
31. Remus, R.: Domain adaptation using domain similarity- and domain complexity-based instance selection for cross-domain sentiment analysis. In: Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012) Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction (SENTIRE). (2012) 717–723
32. Daumé III, H., Kumar, A., Saha, A.: Frustratingly easy semi-supervised domain adaptation. In: Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing (DANLP). (2010) 53–59
33. Daumé III, H., Kumar, A., Saha, A.: Co-regularization based semi-supervised domain adaptation. In: Proceedings of Neural Information Processing Systems (NIPS). (2010) 256–263
34. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL). (2005) 115–124
35. Nakagawa, T., Inui, K., Kurohashi, S.: Dependency tree-based sentiment classification using crfs with hidden variables. In: Proccedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology. (2010) 786–794
36. Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., Manning, C.D.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the 16th Conference on Empirical Methods in Natural Language Processing (EMNLP). (2011) 151–161
37. Wang, S., Manning, C.D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL). (2012) 90–94
38. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th International Conference on World Wide Web (WWW). (2010) 751–760
39. Ponomareva, N., Thelwall, M.: Do neighbours help? an exploration of graph-based algorithms for cross-domain sentiment classification. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL). (2012) 655–665
40. Polanyi, L., Zaenen, A.: Contextual valence shifters. In Shanahan, J.G., Qu, Y., Wiebe, J., eds.: Computing Attitude and Affect in Text: Theory and Application. Volume 20 of The Information Retrieval Series. Computing Attitude and Affect in Text: Theory and Applications. Springer, Dordrecht (2006) 1–9
41. Lakoff, G.: Hedging: A study in media criteria and the logic of fuzzy concepts. Journal of Philosophical Logic **2** (1973) 458–508
42. Reyes, A., Rosso, P.: On the difficulty of automatically detecting irony: Beyond a simple case of negation. Knowledge and Information Systems (2013) 1–20