Extracting word lists for domain-specific implicit opinions from corpora

Núria Bertomeu Castelló
UFS Cognitive Sciences
Universität Potsdam, Germany
parlamind GmbH, Berlin
nuria@parlamind.com

Manfred Stede
UFS Cognitive Sciences
Universität Potsdam, Germany
stede@uni-potsdam.de

Abstract

Sentiment analysis relies to a large extent on lexical resources. While lists of words bearing a context-independent evaluative polarity ('great', 'bad') are available for many languages now, the automatic extraction of domain-specific evaluative vocabulary still needs attention. This holds especially for implicit opinions or so-called polar facts. In our work, we focus on German and on a genre that has not received much attention yet: customer emails. As the prime downstream application is identifying customers' complaints, we concentrate here on finding negative words, but our method applies to positive ones as well. Using a seed list approach, we provide a comparative analysis along three dimensions: effect of different seed lists, different linguistic analysis units, and different statistical correlation tests.

1 Introduction

One interesting and difficult subtask of sentiment analysis is the automatic recognition of so-called *implicit opinions* or *polar facts*: Statements that express a valuation yet do not include context-independent polar words that belong in standard sentiment (polarity) dictionaries. With a polar fact, an author gives a description of some state of affairs, which prima facie appears to be an objective statement, but for the particular target at hand (or more precisely, all targets of its class) entails a polar opinion. They have been studied for product reviews (Toprak et al., 2010) and minutes of meetings (Wilson, 2008), but are also relevant in many other genres such as political or legal discourse.

By their nature, polar fact expressions are domain-specific (see, e.g., Blitzer et al. (2007)). Therefore, it is important to be able to acquire the vocabulary for a domain when high-quality sentiment analysis is to be applied. In this paper, we provide a comparison of several variants of a seed-list approach to generating lists of such lexical items. The starting point is a list of standard opinion words, which we use as seeds to extract collocating polar fact words and phrases from their contexts. The genre we tackle is customer emails, and our data comes from four different content domains, which we illustrate with examples:

- Fashion: "The seam's coming undone."
- Food: "Those cookies were really hard."
- Beauty: "The perfume does not smell!"
- Eyewear: "With these lenses my sight is blurry."

Not surprisingly, most of such customer emails are negative rather than positive: People usually write to the vendor when there is something to complain about. Being able to automatically identify such messages in a company's email stream is an important downstream application for the task we

study here. For that reason, in this paper we will focus on extracting negative polar lexical items; the method, however, would apply to positive items as well.

Our target language is German, but the method is language-neutral, and furthermore it should be applicable to other genres and domains, too. So far there is only little related work on polar fact identification in general, and we are not aware of any that has addressed German. The central aims of our study are to investigate the effects of two different seed sets (varying in origin and size), and of different notions of "minimal unit" for defining the collocation context, and to compare the utility of different measures for lexical association.

The following section summarizes the relevant previous work, and Section 3 introduces our data set. Then, Section 4 presents our experiments and results on word list generation, and Section 5 provides conclusions and an outlook on the next steps of this project.

2 Related Work

Our goal is related on the one hand to the SemEval shared task 2a of 2013 (re-run in 2014 and 2015), where Twitter messages with marked words were given, and the words had to be classified for their polarity. It also resembles SemEval shared task 9 of 2015 on detecting the implicit polarity of events. In contrast to both, however, we are aiming at extracting domain-specific lists of lexical and phrasal items, which can then be applied to the task of finding polar facts. In the following, we thus review previous research on these two aspects: the automatic recognition of polar facts and the generation of word lists for purposes of sentiment analysis.

2.1 Recognizing polar facts

The importance of polar facts has been acknowledged in schemes for manual sentiment annotation by Wilson (2008), Toprak et al. (2010) and de Kauter et al. (2015). There is, however, very little work so far on identifying polar facts automatically. Chen and Chen (2016) extract them from Chinese hotel reviews, but their emphasis is on the additional problem of implicit aspects. Their basic idea is similar to ours in this paper: They collect two adjacent segments, where one contains an explicit opinion word (and an aspect term), and the second one does not, and then project the explicit opinion from one to the other segment (making the assumption that it has the same polarity). New words are learned via the chi-squared test and PMI methods, and candidates are manually inspected for the construction of an opinion word list; the accuracy observed was 70.46%.

2.2 Generating word lists

On the side of *explicit* sentiment, the idea of generating word lists based on a seed set has been employed quite often, and we mention here the successful system of Mohammad et al. (2013); Kiritchenko et al. (2014) for Twitter sentiment classification. Working on two domains, the authors gathered large amounts of unlabelled web data (reviews) and extracted word lists from them. Following Turney and Littman (2003), each term received a sentiment score:

$$score(w) = PMI(w, pos) - PMI(w.neg) \tag{1}$$

with pos and neg denoting positive and negative reviews, respectively. PMI is calculated as:

$$PMI(w, neg) = log_2 \frac{freq(w, neg) * N}{freq(w) * freq(neg)} \tag{2}$$

with $freq(w, neg)$ the number of times a word appears in negative reviews, $freq(w)$ its total frequency in the corpus, $freq(neg)$ the total number of tokens in negative reviews, and N the total number of tokens in the corpus. Terms that occurred less than five times in the positive or negative reviews were ignored. When $PMI(w, pos)$ is calculated in the analogous way, a positive $score(w)$ indicates that w is more associated with positive sentiment, and a negative one indicates negative sentiment. A simple heuristic

negation score recognizer was used to make sure that negated words were treated as different items. The resulting word lists were used in conjunction with other lexicons and contributed significantly to the performance in the overall task of computing aspect-based sentiment (lexicons as a whole increased the F-score by 8 points).

Participating in the same SemEval tasks on Twitter sentiment, Severyn and Moschitti (2015) use a distant supervision method: A large Twitter corpus with noisy polarity labels (inferred from hashtags and emoticons) is mapped to a lexicon of labeled unigrams and bigrams from those Tweets. Then an SVM classifier is trained on these lexical features. The authors show that their method outperforms the PMI-induced lexicon method of Kiritchenko et al. (2014) on the same datasets. Similarly, Vo and Zhang (2016) show that a prediction-based neural network implementation yields a better accuracy than the counting-based method of Mohammad et al. (2013) when comparing to an existing (manually-annotated) "gold" lexicon.

For German, Sidarenka and Stede (2016) compare two families of methods for generating sentiment lexicons and evaluate them on Twitter data: dictionary-based methods starting from WordNet, and corpus-based methods, including the two mentioned above. They found that dictionary-based methods generally outperform corpus-based ones, and that – more importantly for our purposes here – the results of corpus methods depend heavily on the specific seed sets employed in the various approaches. One of the German lexicons available today, SentiWS (Remus et al., 2010), used a method similar to ours – counting co-occurrences, applying log likelihood as association measure, and involving human judgements – and achieved an accuracy of 49.5% for negative polarity words.

3 Our data: Email corpus

Customer care emails are a genre that has not been studied much in the literature, but is highly relevant for several practical applications, including topic identification, sentiment or emotion detection, and automatic mail response systems. Polar facts play an interesting role in these texts, because customers often point out that a product did not meet their expectations, without using explicitly-polar words (see the examples in Section 1). In order not to tie our work too closely to a particular domain, we selected different clients of parlamind GmbH (a company providing customer service solutions) as sources for our email corpus. In a first step we filtered for those emails that actually mention a product, using the Google product taxonomy[1]. The main reason for only selecting e-mails about products is that products are domain-specific (there are e.g. fashion products, eyewear products, beauty products, etc.), while other contact reasons in customer-care e-mails are general accross all the domains (e.g. payment, shipping, withdrawal). Since this work is concerned with finding domain-specific polar words or expressions, we select e-mails with product-related contact reasons. Below we give the number of mails selected by this filter, the sizes of the original sets, and the proportion of the selection.[2] An example email, together with its English translation, is given in Figure 1.

- Fashion (2 clients): 18.358/203.085 emails (9%)
- Food (2 clients): 5.519/35.712 emails (15%)
- Beauty (1 client): 12.819/131.980 emails (10%)
- Eyewear (1 client): 2.087/5.406 emails (39%)
- TOTAL: 38.783/376.183 (10%)

[1] https://support.google.com/merchants/answer/6324436?hl=de

[2] Mails that do no mention a product contain general feedback to the company or its service, ask questions not related to a product, or can be qualified as spam.

Sehr geehrte Damen und Herren,
ich habe am Freitag die Sonnenbrille, die ich bestellt hatte, erhalten. Jedoch fand ich, dass diese Brille nicht gut genug für den Versand verpackt war. Beim Aufsetzen der Brille hat sich mein Gedanke dann bestätigt. Sie ist schief und ist wirklich sehr locker beim Aufsetzen. Sie sitzt so, als wäre sie total ausgeleiert.
Deshalb will ich sie gerne zurück schicken und eine neue, richtig eingepackte Brille zugeschickt bekommen.
Mit freundlichen Grüßen, NAME

Dear Sir or Madam,
on Friday I received the sunglasses that I had ordered, but I noticed that the glasses were not packed up well enough for shipping. When wearing the glasses, this impression was confirmed. They are bended and really very loose when putting on. It feels like they are completely worn out.
Therefore I would like to return them and have new, adequately-packaged sunglasses being shipped to me.
Yours sincerely, NAME

Figure 1: Sample email from 'Eyewear' domain

4 Experiments on word list generation

The system described below was implemented for both positive and negative words, but as explained earlier, we will focus here on the negative set and provide evaluations only for that. The basis for our approach is a seed list of negative words, and we experiment with two variants here: The first is a list of 170 domain-independent negative German words (lemmas) that we compiled manually from various sources, targeting specifically the customer-care email genre. Some translated examples from this list are: *unsatisfactory, unpleasant, dirty, pointless, weak, fault, unfortunately, unreliable, sad.*

The second was obtained from automatically computing the intersection of negative entries in three existing German sentiment lexicons, see (Sidarenka and Stede, 2016). It consists of 9004 words. Henceforth, we abbreviate these lists as NEG-170 and NEG-INTER, respectively.

For computing the polarity of segments, we also use a list of manually-compiled of positive words, POS-100.
The central goal of the experiments is to assess the influence of

- two different ways of obtaining a seed list for negative words,
- different notions of "minimal unit" for the polarity analysis (in the related work above, these were always complete Tweets; we need a more elaborate definition), and
- different measures for computing lexical association.

4.1 Preprocessing

Our pipeline starts with segmenting the emails into minimal units that define the context for computing lexical association between candidate words/phrases and polarity. As there is no generally established definition of that unit, we experimented with four different variants:

- paragraphs (as determined by the newline character)
- sentences (as determined by punctuation, i.e., via sentence splitting)
- discourse units comprising a single or multiple clauses which exhibit discourse continuity among each other. We use discourse markers expressing contrast, such as "aber", "obwohl", "sondern" ("but", "although", "but rather") etc. in order to split mails into discourse units. We perform the splitting in three ways, depending on the type of discourse connective:

  - In the presence of a coordinating discourse marker expressing contrast, such as ”aber”, ”sondern”, ”doch”, ”andererseits” (”but, ”but rather”, ”however”, ”on the other hand”) etc., the text is split and a new discourse unit is created, starting with the marker.
  - In the presence of a subordinating discourse marker, such as ”obgleich”, ”ausser”, ”abgesehen von” (”despite, ”besides”, ”disregarding”) etc., the text is split and a new unit is created that starts with the marker and ends with the next punctuation sign encountered. After that a new discourse unit starts.
  - Finally, if an utterance contains the adverb ”leider”, we create a new discourse unit starting at the beginning of the utterance.[3]

- clauses: main clauses and subordinate clauses are split according to coordinating and subordinating conjunctions and punctuation (final punctuation or comma occurring after a finite verb). Infinitive clauses, complement clauses, irrealis conditional clauses[4] and indirect questions are not split because the main clause has incomplete meaning without the subordinated clause.

The example in Figure 1 consists of four paragraphs, most of them very short; the second gets segmented into five sentences. In turn, two of these sentences get split into two clauses. The e-mail contains two discourse units.

Generally, sentences that are questions get removed from the corpus, as we do not expect to find opinions in there. For POS tagging, we use the Apache OpenNLP[5] tagger with its German model, but performed additional training on a manually-annotated email corpus, in order to improve the performance on our genre. Based on the POS tags, we then mark the candidate items that will be considered for our lexicon: nouns, adjectives, and verbs. In addition, we extract predicate/argument tuples: For each verb, we build a set of tuples containing all combinations of the verb with the respective heads of the subject, object and indirect object. I.e., for a transitive verb, as in *We process the order*, three such tuples are being built: *(process,we), (process,order)* and *(process,we,order)*. The tuples are extracted from predicate-argument structures. Those are obtained using chunking heuristics and morphological information to identify the syntactic roles in the sentence. When it is unambiguous, morphological case informs the selection of syntactic roles, otherwise constituent-order is used. This can be considered a “light” version of dependency analysis (full parsing has generally turned out to be too noisy on the email data).

As part of the predicate/argument analysis, we check for the presence of negation operators, i.e., words such as ”nicht”, ”kein”, ”keins”, ”niemand”, ”nie”, ”nirgendwo” (”not”, ”no”, ”none”, ”noone”, ”never”, ”nowhere”) etc. and heuristically determine their scope: negated verbs and negated arguments are stored as such. If a clause contains some negated argument, it is stored as negated, too. Therefore if a tuple contains any negation, it is stored as negated.

We then mark all instances of negative and positive words (from our seed lists) in all the segments and add their polarity as a feature. Again, we check for negation scope and reverse the polarity if necessary. Based on these assignments, the final step is to label each segment with its estimated polarity:

- positive: If there is a majority of positive instances (having factored in the negation scope);
- negative: (likewise);
- neutral: if the number of positive and negative words is equal.

[3]This decision is based on the observation that customer-care e-mails expressing complaints very often start with a small narrative introduction and then signal the start of the complaint with an adverb such as ”unfortunately”. For example: ”On Friday the shirt that I had ordered arrived. Unfortunately, the shirt is too small for me.” Here, the adverb ”unfortunately” behaves similarly to a discourse marker like ”but”.

[4]For example: ”Es wäre schön, wenn Sie mir das noch mal zuschicken könnten.” (”It would be nice if you could send it to me again.”)

[5]https://opennlp.apache.org

Polarity	Item	Not Item	Total
-1	6 (c12)	19172	19178 (c1)
0,1	0	146699	146699
Total	6 (c2)	165871	165877 (n)

Table 1: Sample contingency table, as built for each candidate item (word, pred/arg tuple)

At the end of this preprocessing phase, we have for each of the four domains a set of segments (coming in four variants, as described above) with a polarity label and all potential candidate items (word, pred/arg tuples) included therein.

4.2 Scoring and ranking the candidates

For building our different variants of domain-specific lexicons, we now consider each candidate item and count how often it occurs in a positive, negative, and neutral segment. For building a list of negative items (which is the goal scenario for the rest of the paper), we compute a score for each item, following six different methods. Similar to the related work, we remove items that occur less than three times in all the negative segments. Then, for each of the scoring methods, a ranked lists of the items is produced. (Recall that this complete process is also run for each of the above-mentioned notions of elementary unit.) In the following, we describe the scoring methods, based on the sample contingency table shown in Table 1. (Such a table is constructed for each word/item under consideration.) It uses the following variables:

- c12: co-occurrence of the item and the currently-considered polarity
- c1: total counts of the considered polarity
- c2: total counts of the considered item
- n: total number of units (segments)

Bayesian test We use two binary variables: pol = polarity (1 = negative; 0 = not negative) and i = item (1 = present; 0 = not present). Then we estimate the probability that the polarity is negative, given that we have observed the item:

$$P(pol = 1|i = 1) = k * (P(pol = 1) * P(i = 1|pol = 1)) = (c1/n) * (c12/c1) \quad (3)$$

where k is a normalizing constant, so that $P(pol = 1|i = 1) + P(pol = 0|i = 1) = 1$.

Frequency classes difference test Here we consider that we have two (sub-)corpora: one with segments of the negative polarity under consideration and one with other segments. The approach is for a given item to find out whether it belongs to a different frequency class in each of the corpora, and if so, how distant those frequency classes are. The lower the frequency class, the more frequent is the word in the given corpus.

- Polarity-Frequency-Class: $log_2(c1/c12)$
- Reference-Frequency-Class: $log_2((n - c1)/(c2 - c12))$
- Difference: $log_2(c1/c12) - log_2((n - c1)/(c2 - c12))$

Relative likelihood ratio test The relative likelihood ratio test assumes the same two sub-corpora. It is the ratio of the probability of seeing an item given the negative polarity over the probability of seeing the same item given the non-negative polarity. It is calculated as follows:

$$\frac{c12/c1}{(c2-c12)/(n-c1)} \tag{4}$$

Likelihood ratio test We use the same binary variables pol and i as in the Bayesian test. Then we calculate the ratio of the null hypothesis (a given item and a given polarity are independent) over the alternative hypothesis (the given item and polarity are dependent):

- H1 (null hypothesis) $= P(pol = 1|i = 1) = p = P(pol = 1|i = 0)$
- H2 (alternative hypothesis) $= P(pol = 1|i = 1) = p1 \neq p2 = P(pol = 1|i = 0)$
- $log(L(H1)/L(H2)) = logL(c12, c2, p) + logL(c1 - c12, N - c2, p) - logL(c12, c2, p1) - logL(c1 - c12, N - c2, p2)$, where
  - $L(k, n, x) = x^k(1 - x)^{(n-k)}$
  - $p = c1/n$
  - $p1 = c12/c2$
  - $p2 = (c1 - c12)/(N - c2)$

Chi-squared test The Chi-squared test sums the differences between observed and expected values in all squares of the table, scaled by the magnitues of the expected values.

$$\frac{n * ((o11 * o22) - (o12 * o21))^2}{(o11 + o12) * (o11 + o21) * (o12 + o22) * (o21 + o22)} \tag{5}$$

where

- $o11 = c12$
- $o12 = c1 - c12$
- $o21 = c2 - c12$
- $o22 = n - c12 - o21$

PMI test

$$log_2 \frac{c12/n}{(c1/n) * (c2/n)} \tag{6}$$

From each of the domain-specific ranked lists of items (as they result from the different combinations of segmentation units and statistical tests), we take the top 350 items. The 350 top words cut-off is motivated both by the amount of items our annotators could manage and by the perceived lower (negative) quality of the words appearing after the 350 cut-off. This perceived lower quality is later confirmed by the probability of 0.57 of being negative in the Bayesian test of the words occurring after the threshold and the precision of 0.528 on the cut-off. See section 4.3 for evaluation scores on different thresholds.

In the following, we present the procedure and evaluation for the Fashion domain, as it has the largest amount of data. Here, the union of the sets of 350 words amounts to 2146 unique items. For these, we obtained human judgements as to whether they are indeed negative, given the domain in question. (For example, in the Fashion domain, the tuple (have,hole) would be judged as a proper negative item.) Our annotators confirmed that 559 of the 2146 items are negative (in their respective domain).[6]

These 559 "gold" items are the basis for the evaluation of the various settings we are interested in: choice of seed list, type of minimal unit, and association measure.

[6] At this point we did not compute inter-annotator agreement for the polarity assignments; this is left for future work.

Bayesian	Chi-squared
(reklamiren,Mangel) *(complain,fault)*	klein *small*
(unterlaufen,Box) *(occur,box)*	feststellen *notice*
(haben,Mangel) *(have,fault)*	eng *tight*
(zeigen,Mangel) *(show,fault)*	muss *must*
(feststellen,Rücksendung) *(notice,return shipping)*	musste *had to*
(setzen,Defekt) *(sit,defect)*	groß *large*
(erwarten,Monat) *(expect,month)*	(haben,Problem) *(have,problem)*
(scheinen,Fehler) *(appear,mistake)*	unterlaufen *occur*
leiert *worn out*	weit *wide*
(haben,Defekt) *(have,defect)*	Qualität *quality*
(verändern,Farbe) *(change,color)*	(unterlaufen,Fehler) *(occur,mistake)*
(entstehen,Problem) *(originate,problem)*	technisch *technical*

Table 2: Top items of the generated "Fashion" term list (negative) for two methods

4.3 Evaluation and results

For each ranked list of negative items as produced by one of the settings, we again consider the top 350 items and measure how many of them are indeed negative (i.e., they are among the 559 confirmed ones). We determined these accuracy scores for the Fashion domain. For illustration, Table 2 shows the top 12 entries in the lists of the extracted negative items for two association measures. Evidently they differ in favouring single words versus tuples, which we comment on below. The verb 'unterlaufen' *happen* in German collocates predominantly with negative event nouns ('Fehler', *mistake*) but is not by itself negative. Thus, the items in the Bayesian list indeed all indicate negative sentiment; two thirds of them contain a generally-negative word such as *problem*, while one third represent domain-specific polar facts (e.g., *(change,color)*). The Chi-squared list, on the other hand, has several items that cannot be identified as negative without knowing further context (e.g., *small, large*). We surmise that they are often used with the intensifier 'zu' *too*, which then yiels a negative judgement. The modal verb 'müssen' *must* in a customer email has a negative ring, similar to 'unterlaufen' *occur*, which is present in this list as well.

Table 3 contains twenty top negative words for the domains Fashion and Eyewear obtained with the Bayesian method, all of them with probability 1.0 of being negative. Tuples containing generally-negative words, such as *problem*, have been filtered out for presentation. As you can see, both lists contain genre-specific domain-independent words and tuples (such as "(notice, return shipping)"or "(transfer, bill"), but also domain-specific or at least domain-relevant words and tuples (such as "worn out" and "loosened", for the Fashion domain, and "exact-FALSE"[7] and "(have, assembly, glasses)-FALSE", for the Eyewear domain). Many tuples and words are not negative per se, but they appear as negative because in the customer-care-genre and the specific domains they are only mentioned in relation to some trouble or issue; otherwise they are seldom the topic of some mail (e.g. "loose money pocket"or "water-repellent")[8]. So, even if this kind of words are not polar, it may be helpful to consider them for the purpose of identifying complaints.

Seed set. We consistently obtained much better results for NEG-170 than for NEG-INTER. For example, the overall best result (any unit, any association measure) for NEG-INTER is 0.35, while for NEG-170 we often get results over 0.5, as can be seen in Table 4. We assume that the long, automatically-generated NEG-INTER list has too much noise and produces many irrelevant results for our type of task. Another possibility is that NEG-INTER does not contain some genre-specific sentiment words that occur particularly often in customer-care e-mails (e.g. "Schrott" (*scrap*), "unerklärlich" (*unexplicable*)). This

[7]The suffix "-FALSE" is to be understood as negation, e.g. "not exact".

[8]Please take into account that orders are always taken automatically in the online-shop and only very seldom per e-mail. So those domain-relevant words are usually only mentioned in the context of complaints.

Fashion	Eyewear
(feststellen,Rücksendung) *(notice,return shipping)*	Paketversand *packet shipping*
(unterlaufen,Box) *(occur,box)*	(unterlaufen, Bestellung) *(occur, order)*
leiert *worn out*	erfolglos *unsuccessful*
(haben, Etikett) *(have, label)*	(erhalten, Rückmeldung)-FALSE *(receive, response)-FALSE*
(zeigen,Herstellerlabel) *(show, producer label)*	abhanden *lost*
Frontseite *front side*	(haben, Fertigung)-FALSE *(have, assembly)-FALSE*
(vermeiden, Rücklastschriftgebühren, Mahnkosten) *(avoid, return debit note fee, dunning costs)*	Rücksendung-FALSE *return-FALSE*
DHL-Lieferantin *DHL-delivery woman*	(durchführen, Messung) *(take, measurements)*
(angeben, Hosengrösse) *(provide, trousers size)*	genau-FALSE *exact-FALSE*
erwischen *to catch*	angerechnet *charged*
(zusenden, bitte, Beutel) *(send, please, bag)*	Ring *ring*
loesten *loosened*	herausgestellt *turned out*
erst-FALSE *first-FALSE*	(haben, Fertigung, Brille)-FALSE *(have, assembly, glasses)-FALSE*
reklamiren *to complain*	wasserabweisend *water-repellent*
Retoureschein-FALSE *return voucher-FALSE*	Zustand *state*
(gutgeschrieben, Herr, Betrag, Rechnung)-FALSE *(booked, Mister, amount, bill)-FALSE*	(verlegen, Rechnung) *(transfer, bill)*
begangen *commited (e.g. error)*	(gelten, Rücksendung)-FALSE *(be valid, return)-FALSE*
Paketanfrage *packet inquiry*	schicht *coating*
Kleingeldfach *loose money pocket*	laufen-FALSE *work-FALSE*
(verwundern, Situation) *(wonder, situation)*	spät *late*

Table 3: Top items of the generated “Fashion” and ”Eyewear” term lists (negative) without tuples containing seeds

Segmentation	rel. likel. ratio	freq. class	log likel.	chi-squared	Bayesian	PMI
clauses	0.53	0.53	0.36	0.41	0.53	0.53
sentences	0.52	0.52	0.31	0.31	0.52	0.52
disc. units	0.46	0.46	0.25	0.23	0.46	0.46
paragraphs	0.40	0.41	0.21	0.19	0.41	0.41

Table 4: Results (accuracy) for the NEG-170 seed set, Fashion domain

suggests that having genre-specific seed words (in this case, customer-care sentiment words) may be more useful to obtain domain-specific lists of words (e.g. for Fashion, Eyewear, Beauty, ...) than using general sentiment seeds.

Unit. In Table 4, and in our other experiments, we find that clauses are generally the most successful unit of analysis for our task. This can be related to the genre of the customer emails, which very often are a mix of neutral reporting ("I received the ordered box last week", etc.) and a single specific complaint. Sometimes, this complaint is actually part of a contrastive sentence ("I like the color of the blouse, but it really doesn't fit", etc.). This is in contrast to text-level sentiment analysis, as needed for product or movie reviews, where the general tone of the overall text is to be determined.

Association measure. Overall, the relative likelihood ratio test, the frequency classes difference test, the Bayesian test and the PMI test achieve the best results. The fact that the log likelihood and Chi-square tests turn out considerably worse might be a result of our setting where tuples play an important role and have generally a better chance to get a positive vote from the human annotators; in particular, a verb-object combination is often more easily judged as negative in the domain than just the single verb. The two last-mentioned tests, however, yield more individual words than the others do, because the words occur more frequently than the tuples, and the two tests are more sensitive to this than the others are (which are based on probabilities independent of the absolute frequency in the corpus). In general, the other tests rank higher items occurring only in one polarity, independently of their frequency.

Overall best results. Besides accuracy, we also measured precision, recall, and F1 for the various combinations of settings that we investigated, and with various thresholds. The maximum F1 score we obtained is 0.55 (for a threshold of 0.742, using the Bayesian measure), with precision at 0.65 and recall at 0.48. The optimal precision is (also using the Bayesian measure, with a threshold of 0.957) 0.73, accompanied by a recall of 0.4.

4.4 Comparison to a baseline

In order to substantiate the findings on unit size and on the influence of our preprocessing, we also implemented a baseline that follows the approach of Turney and Littman (2003): There is no segmentation; instead, for each seed word found, we look for collocation candidates within a fixed window of ten words to the left and to the right of the seed word. We do not identify negation operators and compute any associated polarity reversals. Only single words are being considered, no predicate/argument tuples. The correlation measure is PMI, but with counts replacing probabilities (as also done by Turney and Littman).

For the setting with NEG-170, as reported in Table 4, this methods yields an accuracy of only 0.09, which is substantially worse than the PMI results for the other units, which include the preprocessing measures.

5 Conclusion

Customer emails are a genre that invites many practical applications involving sorting the mails according to the presumed intention of the sender. Negative sentiment in a mail tends to indicate a complaint, and

is thus an important feature for a processing pipeline. Recognizing the sentiment, however, is to a large extent dependent on the ability to recognize implicit opinions (or 'polar facts'), which is in turn a domain-dependent task. In the experiments reported here, we worked with German mails from four different domains and selected 'Fashion' as the one to run the evaluations for.

The basic idea we employ here, using domain-neutral seed words to identify further domain-specific negative items, is not new. But we constructed a number of variants of the task, in order to determine some crucial features for a working solution. In our experiments, we thus combined one of two seed lists, one of four notions of minimal unit for allocating candidate items, and one of six different lexical association measures, in order to find the overall most promising combination. In the absence of a large amount of test data, we used human judgements of interim results (the union of the top 350 items suggested by the best-performing combinations) in order to produce the confirmed lexical items (many of them domain-specific) that were then used to evaluate the approach. Although not directly comparable, because different corpora are used, our method, with accuracies of 0.53, seems to be competitive to the earlier work on generating German negative words by Remus et al. (2010) (see Section 2), which achieved an accuracy of 0.495.

A further difference to earlier work is our use of tuples of verb and dependents in addition to individual lexical tokens (or bigrams). These tuples represent a step toward syntactically-motivated analysis while avoiding the full parsing problem (which we found to be quite hard for the email genre). The lists generated by our best-performing settings contain many such tuples, which indicates the potential of this approach. Further improving this minimal syntactic analysis (e.g., by handling verb suffixes) is one of our goals for the future work.

To our knowledge, the influence of using a specific notion of minimal unit for the computation of lexical associations with seed words has not beed studied before. For one thing, much of the earlier work used token windows of a fixed size; in addition, much work has been done on Tweets, which were taken as a complete unit of analysis. For emails, experimenting with units of different size is an important step, and we found that (heuristically-determined) clauses are the best choice, outperforming discourse units (also computed heuristically, drawing on the presence of connectives) and larger units (sentences, paragraphs).

Furthermore, we found that a carefully-selected genre-specific seed list of 170 negative words significantly outperforms a list that is derived from the intersection of three widely-used German sentiment lexicons; these have been built with semi-automatic methods, and apparently contain too much noise for the task at hand here and lack relevant sentiment words in the customer-care e-mail genre.

While the association measures have some diverging properties and are therefore not straightforwardly comparable, we found that generally, the chi-squared and log likelihood tests performed worse than the four others we had implemented.

In order to develop our findings further toward a practical procedure for determining domain-specific negative items, one important task is to empirically motivate a threshold for the ranked list of generated items, which is taken to separate the items regarded as "trustworthy" from those that are not. So far, we used a fixed 350-item cutoff for our experiments, motivated in part by the volume that could be handled by our human judges in dis/confirming the suggestions by the algorithm and by our initial intuitions regarding the quality of the items according to their rank. Our ongoing experiments with precision/recall measurements (some of which we mention in Section 4) are a step into this direction, which constitutes a major aim of our future work.

Acknowledgments

The work reported here was financially supported by the German Federal Ministry for Education and Research (BMBF) through grant 03EFFBB037 in the 'EXIST Forschungstransfer' programme. The authors thank Uladzimir Sidarenka for helpful comments on earlier versions of the paper, and the anonymous reviewers for their constructive suggestions.

References

Blitzer, J., M. Dredze, and F. Pereira (2007). Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 440–447. Association for Computational Linguistics.

Chen, H.-Y. and H.-H. Chen (2016). Implicit polarity and implicit aspect recognition in opinion mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, pp. 20–25. Association for Computational Linguistics.

de Kauter, M. V., B. Desmet, and V. Hoste (2015). Guidelines for the fine-grained analysis of polar expressions. Technical Report LT3 14-02, Ghent University.

Kiritchenko, S., X. Zhu, C. Cherry, and S. Mohammad (2014). NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Dublin, Ireland, pp. 437–442. Association for Computational Linguistics and Dublin City University.

Mohammad, S., S. Kiritchenko, and X. Zhu (2013). 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 Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, pp. 321–327. Association for Computational Linguistics.

Remus, R., U. Quasthoff, and G. Heyer (2010). SentiWS - a publicly available German-language resource for sentiment analysis. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Severyn, A. and A. Moschitti (2015). On the automatic learning of sentiment lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, pp. 1397–1402. Association for Computational Linguistics.

Sidarenka, U. and M. Stede (2016). Generating sentiment lexicons for German Twitter. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES) at COLING*, Osaka, Japan, pp. 80–90.

Toprak, C., N. Jakob, and I. Gurevych (2010). Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 575–584. Association for Computational Linguistics.

Turney, P. and M. L. Littman (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems 21*(4).

Vo, D. T. and Y. Zhang (2016). Don't count, predict! an automatic approach to learning sentiment lexicons for short text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, pp. 219–224. Association for Computational Linguistics.

Wilson, T. (2008). Annotating subjective content in meetings. In *Proceedings of the 6th conference on language resources and evaluation (LREC)*, pp. 2738–2745.