ACCEPT

SEVENTH FRAMEWORK PROGRAMME

THEME ICT-2011.4.2(a)

Language Technologies

ACCEPT

Automated Community Content Editing PorTal

www.accept-project.eu

Starting date of the project: 1 January 2012

Overall duration of the project: 36 months

Rules for automatic sentiment detection and report of evaluation results

Workpackage n° 3	Name: Text Analytics
Deliverable n° 3.4	Name: Rules for automatic sentiment detection and report of evaluation results
Due date: 31 December 2014	Submission date: 19 December 2014

Dissemination level: PU

Organisation name of lead contractor for this deliverable: Acrolinx

Authors: Christian Bering, Robert Grabowski, Johann Roturier

Internal reviewers: Pierrette Bouillon, Johanna Gerlach, John Papaioannou, Johann Roturier

Proofreading: Emmanuel Rayner

Copyediting: Silvia Rodríguez Vázquez

The research leading to these results has received funding from the European Community's Seventh Framework Programme (*FP7/2007-2013*) under *grant agreement n° 288769.*

SEVENTH FRAMEWORK PROGRAMME

Contents

List of Figures

List of Tables

Rules for automatic sentiment detection and report of evaluation results

Foreword

As agreed with the Project Officer, the original deliverables D3.3: *Rules for automatic sentiment detection* and D3.4: *Report of evaluation results* are being merged into the present, common deliverable (D3.4).

1 Rules for Automatic Sentiment Detection

1.1 Objective

The main goal of this first part of the deliverable is to present the Acrolinx rules developed in Task 3.3 to automatically check for content sentiment polarity, i.e., whether a given document is heavily charged with positive or negative sentiment. The rule sets have been integrated into the linguistic resources on the hosted Acrolinx server for ACCEPT. They can be selected as "Sentiment-SentiWords" (for English) and "Sentiment-OpeNER" (for French).

1.2 Overview and Structure

This section describes the expected requirements and specifications with regard to sentiment analysis, explains the approach and presents the structure of the following sections.

We concentrated on lexical features; this is in contrast to more complex features such as entity-aspect-relational features (Liu, 2012). Lexical features work well both for document-level and for segment-level analysis. We have developed and tested methods for English and French sentiment analysis. Where applicable, we have accumulated resources for further languages to facilitate future application of successful approaches.

Among the simplest indicators that have been successfully used in sentiment analysis are lexical n-grams (Wang and Manning, 2012). While it has been noted that the adequate selection of lexical sentiment indicators can be difficult for humans (Pang et al., 2002), not to mention unfeasible for real-world applications, supervised machine learning techniques can derive lexical n-gram models efficiently and reliably even from large quantities of pre-classified text. However, a model learnt in this fashion draws its features directly from the vocabulary of the training data and might consequently not generalize well beyond that domain. On the other hand, in its specific domain, it is bound to outperform approaches based on more generic vocabulary.

A complementary approach to learning lexical features from a domain-specific corpus is to use an independently compiled, generic sentiment-annotated lexicon in the respective language. Items from such lexica are ideal building blocks for the linguistic patterns which Acrolinx rules detect and mark (flag) in texts. Previously, we have developed prototypical Acrolinx sentiment rules to detect negative sentiment.

Figure 1 shows an example of morphosyntactic objects from these prototypical Acrolinx sentiment rules which target specific negative expressions using morphological information. The defined objects

demonstrate how patterns can pertain to domain-specific issues (such as software problems in @bug) or more general notions such as amplifiers. Since the rules make use of the syntactic structure of the text, they can be generalized to handle inflections and cope with negation. Figure 2 shows examples of Acrolinx rule patterns which use such linguistically specified objects to mark adjective phrases that convey positive sentiment as well as negated negative, i.e., positive sentiment utterances. The rules are associated with confidence values (shown as 80 in the examples) which are used to attach scores to flags.

```
67
68 @bug     ::=     [ MORPH.LEMMA "^(bug|crash|freeze|nightmare)$"
69                   POS "NN" ];
70
71 @problem    ::=     [ MORPH.LEMMA "^(bug|issue|problem|trouble)$" ];
72
73 @amplifier  ::= [  TOKEN.FORMS { "^(absolutely|ABSOLUTELY|as|AS|extremly|EXTREMELY|hugely|HUGELY|kinda
74
75
76 @predVerb   ::= {[  MORPH.LEMMA "^(appear|be|find|seem)$" ]
77                 |[  TOK "s$"
78                     TOKCLASS "RightSplit" ]};
79
```

Figure 1: Lexical definitions in Acrolinx sentiment resources

```
TRIGGER (80) ==  @amplifier [{@adv_all|@adjective}]*!^1 @keyword_pos_adj^2
                -> ($adjective, $headAdj)
                -> { mark: $adjective, $headAdj;
                }

TRIGGER (80) ==  @negation [{@amplifier|@adv_all|@adjective}]*!^1 @keyword_neg_adj^2
                -> ($adjective, $headAdj)
                -> { mark: $adjective, $headAdj;
                }
```

Figure 2: Example rule patterns for positive sentiment

In Task 3.3, we created a generic sentiment classifier for English based on Acrolinx rules and generic sentiment lexica. This classifier aims to distinguish positively charged from negatively charged texts.

To get an idea of the potential performance of this generic approach in our specific application scenario, we first applied the existing prototypical rules to a corpus of posts by Symantec forum users. Its performance gives a baseline for the new rule-based sentiment classifier. For an upper bound of the rule-base classification performance, we created domain-specific lexical models learnt from the Symantec forum posts as well as externally collected Amazon reviews (for lack of further sentiment-annotated material).

We then generated new Acrolinx sentiment rules using English and French lexica taken from the SentiWord corpus (Guerini et al., 2013) and the OpeNER project[1], respectively. We trained a sentiment classifier based on the rule flag occurrences and evaluated its precision and recall.

The structure of the deliverable is follows: Section 1.3 gives an overview of the employed data sets. Sections 1.4 and 1.5 discuss the baseline and the upper bound experiments, respectively. The

[1] See http://www.opener-project.eu/ (Accessed: November 2014).

generated sentiment rules, the trained classifier based on these rules and its performance are presented in Section 1.6.

1.3 Data Sets

1.3.1 Messages Containing Symantec Mentions

We have processed posts from a range of social media sources (Twitter, user forum, etc.) in English and French, in which users mentioned Symantec products. The posts were then labelled with one of eight nominal content classes using a semi-automatic classification process: The posts were first classified automatically, and the result of that step was manually verified. The classes were defined as follows:

- Case: "Request for help resolving real-time issue"
- Lead: "Pronouncement of near-term purchase decision"
- Query: "Question that doesn't require support resource"
- Rant: "Insult that merits brand management consideration"
- Rave: "Praise from Symantec brand advocate"
- RFE: "Suggestion for product feature or improvement"
- Noise and Fraud: Post unrelated to products.

The English user posts corpus consists of 3643 posts from a range of sources. Overall, the posts are rather short; Table 1 shows the average word lengths by content type, and Table 2 shows the frequencies of the posts by content class and sentiment. Approx. 66.7% are classified as *Noise*, while *Rants* make up for approx. 7% of the posts. For the purpose of classifying positively and negatively charged posts, we assume that *Rants* are negative, while *Raves* would be positive. Part of the baseline experiments was to determine whether any of the other content classes would lend themselves to a similar sentiment interpretation.

In the same fashion, the posts were classified along an ordinal sentiment scale with the levels "Negative" / "Somewhat negative" / "Neutral" / "Mixed" / "Somewhat positive" / "Positive". As Table 2 shows, almost no posts were classified as "Somewhat negative/positive" or "Mixed". This is quite likely due to the fact that the posts were too short to allow for finer distinctions for emotional content. For "Mixed", annotators might have additionally confused it with the "Neutral" classification.

Content type	avg	std
Noise	39.24	20.70
Rave	24.71	16.64
Rant	36.42	19.65
Case	46.30	20.10
Query	37.80	19.03
Fraud	20.92	16.16
Lead	50.29	19.15
RFE	32.69	16.41
Overall	37.29	20.83

Table 1: Average word lengths and standard deviations of English posts by content types

Content type	Total	Negative	Somewhat negative	Mixed	Neutral	Somewhat positive	Positive
Case	234	226	0	0	7	1	0
Fraud	117	0	0	0	117	0	0
Lead	41	5	0	1	35	0	0
Noise	2430	2	0	0	2426	1	1
Query	132	0	0	0	131	0	1
RFE	16	0	0	0	16	0	0
Rant	256	244	3	1	8	0	0
Rave	417	0	0	0	3	0	414

Table 2: Frequencies of English posts by content type and sentiment

The French Symantec corpus contains 2690 unique posts and shows characteristics similar to the English collection. Table 3 shows the distribution of posts by content category and the distribution over sentiment classification.

Content type	Total	Negative	Neutral	Positive
Case	52	52	0	0
Fraud	30	0	30	0
Lead	94	1	93	0
Noise	2181	1	2161	0
Query	24	0	24	0
RFE	1	0	1	0
Rant	135	119	16	0
Rave	173	0	13	159

Table 3: Frequencies of French posts by content type and sentiment ("Mixed" and "Somewhat" sentiment classes with zero posts have been omitted)

1.3.2 Amazon

We have obtained a review corpus of 5.84 million English Amazon reviews from an online sentiment analysis research resource (Jindal and Liu, 2008). The corpus was originally collected for usage in opinion spam (fake review) detection. It contains some duplicates (often minor variations of the same review) and some non-English reviews. With these filtered out, the dataset consists of 5.77 million English reviews, where each review rates an associated product with between one and five stars. The distribution over ratings is highly skewed: 5-star-ratings alone make up for approximately 57% of the reviews.

Adopting the methodology of Jindal and Liu (2008), we have retrieved 1.23 million Amazon reviews in French. Similar to the English corpus, we have pre-processed the collection by removing duplicates and reviews in other languages. Furthermore, we have collected German and Japanese Amazon reviews (approx. 2.51 million and 1.15 million, respectively) in preparation for comparable work in these languages.

1.4 Baseline Tests With Existing Prototypical Rules

In order to establish a baseline for the performance of sentiment rules, we tested generic, manually generated Acrolinx sentiment rules (similar to rules depicted in Figure 1 and Figure 2) on the corpus of English Symantec user forum posts. An important question would be whether the rules' flag scores allow to distinguish between posts from different content classes. However, since sentiment

expressions can depend on lexical cues specific to application areas, we expected these generic sentiment rules to have a low discriminative power.

Table 4 shows the results of checking the Symantec user posts with the generic Acrolinx rules for negative sentiment. Each row shows the following:

- the content type;
- the number of occurrences of this content type in the corpus (Total);
- the percentage of posts in which no flag was found (0%);
- the maximum sum of flag scores[2] found in a post of this type (max); and
- the average score and standard deviation of summed scores per post (avg, std).

Content type	Total	0%	max	avg	std
Case	234	76,92%	1250	76,78	±176.81
Fraud	117	36,75%	2000	571,63	±490.60
Lead	41	68,29%	667	92,2	±173.99
Noise	2430	69,30%	2857	151,99	±299.91
Query	132	78,79%	833	76,14	±170.42
RFE	16	75,00%	1053	139,62	±290.99
Rant	256	66,02%	1429	151,54	±259.16
Rave	417	36,93%	1429	416,39	±378.68

Table 4: Results for checking Symantec user posts with generic Acrolinx sentiment resources

It can be seen that, with the notable exception of *Fraud* and *Rave*, between 66% and 77% of posts across content classes failed to elicit even a single negative sentiment flag. This also pertains to the assumed negative target class of *Rant*. Similarly, neither maximum nor average scores of markings provide sufficient means to distinguish either *Rants* or *Raves* from other categories. Thus, an important step is to determine which lexical items can distinguish between *Rants* and non-*Rants*.

1.5 Sentiment Analysis using Domain-Specific Vocabulary

In order to determine the lexical base for a sound sentiment rule setup, we trained and tested classifiers in different experimental setups. In a pre-processing step, we lemmatized the posts using Acrolinx' internal Acrotokenize functionality. For each post, we determined the frequencies of lemma uni- and bigrams and trained a Multinomial Naïve Bayes classifier[3] using count-based "bags of words" models. These have been shown to have better discriminative power than binary models, in which the absence or presence of a word in a document is used regardless of frequency (McCallum and Nigam, 1998). Except where noted elsewhere, we filtered out n-grams which occurred only once in the training documents and eliminated English stop words.[4]

For each experimental condition, we ran ten experiments in ten-fold cross-validation: the data set was split into ten partitions with a similar percentage of *Rants* in each. In each experiment, nine partitions

[2] As introduced in Section 1.2.
[3] As provided by the Python-base machine learning package *scikit-learn*, which can be obtained from http://scikit-learn.org (Accessed: December 2014).
[4] Control runs which included stop words yielded no significantly differing results.

(i.e., 90% of the data) were used for training, and the remaining tenth partition (10%) as test data, such that every partition was used as a test partition once.

An important aspect of determining a feasible set of lexical items is to control for the relevant subset of all available n-grams (here: unigrams and bigrams) which still allow for an adequate classification. To achieve this, we introduced a pre-learning phase in which the 90% training data were classified using a decision tree learner ensemble. The resulting model of the decision trees can be used to rank the features by their influence on the classification. Only the most influential subset of features would then be used for the actual training. The selection of the decision tree learners can be used as the lexical vocabulary for building Acrolinx rules.

1.5.1 Amazon Reviews Classification

To test the general validity of the experimental setup, we devised an experiment to separate 1-star Amazon reviews from 5-star reviews. From the English Amazon corpus, we selected 10000 reviews (5000 of these with a 1-star rating, 5000 5-star) pertaining to software products or books about software. In order to have a comparable document size, we only used reviews below 1000 characters.

Figure 3 shows the results of this classification. The three plots on the left show the results for 1-star reviews, the right-hand column shows the corresponding plots for the 5-star reviews. Each column shows precision, recall, and f1, from top to bottom, both for the performance on the training data and on the test data (lighter curve). It can be noted that the overall performance is consistently around and mostly above 80%. Filtering out up to approximately 95% of the n-gram vocabulary has no negative effect on this performance.

Table 5 shows two example sets of the ten lexical items with the highest discriminative power according to the decision tree pre-learning stage in two different runs. Because of the ordering by discriminative power, the sets both contain indicators for both sentiment polarities (e.g., “excellent” as a likely indication of a five-star review; “bad” as a likely indicator for a one-star review).

Run 1	Run 2
error	waste money
great book	poor
easy	highly
waste money	easy
useless	bad
excellent	book
money	waste
great	great
waste	excellent
bad	money

Table 5: Two sets of n-grams from two different pre-learning runs with the highest discriminative weights in decision tree learning for Amazon review classification, sorted by weight (highest first)[5]

[5] Shaded entries appear in both sets.

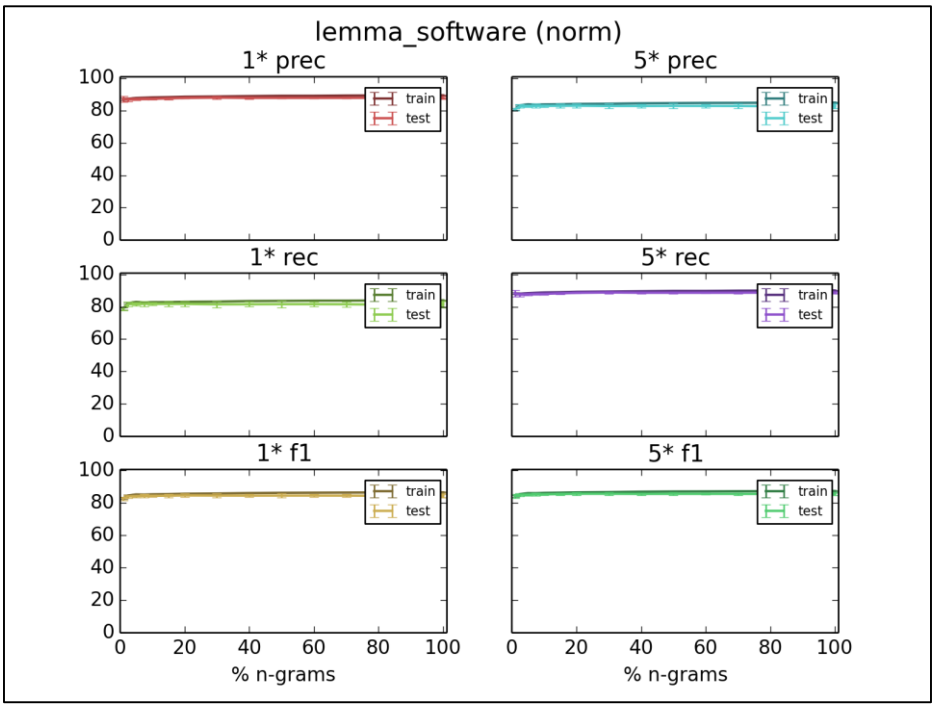

Figure 3: Recall (rec), precision (prec), and f1 measure for classifying 1-star (1*) Amazon software reviews versus 5-star (5*) reviews as a function of the percentage of n-grams in the training documents

1.5.2 Symantec Posts Classification

We applied the same experimental setup to *Rant* discrimination for Symantec user posts. Figure 4 shows the results. While performance was consistently near perfect for non-rants, this comes at a cost for the classification of rants, which only had an accuracy of around 0.3 (f1 score).

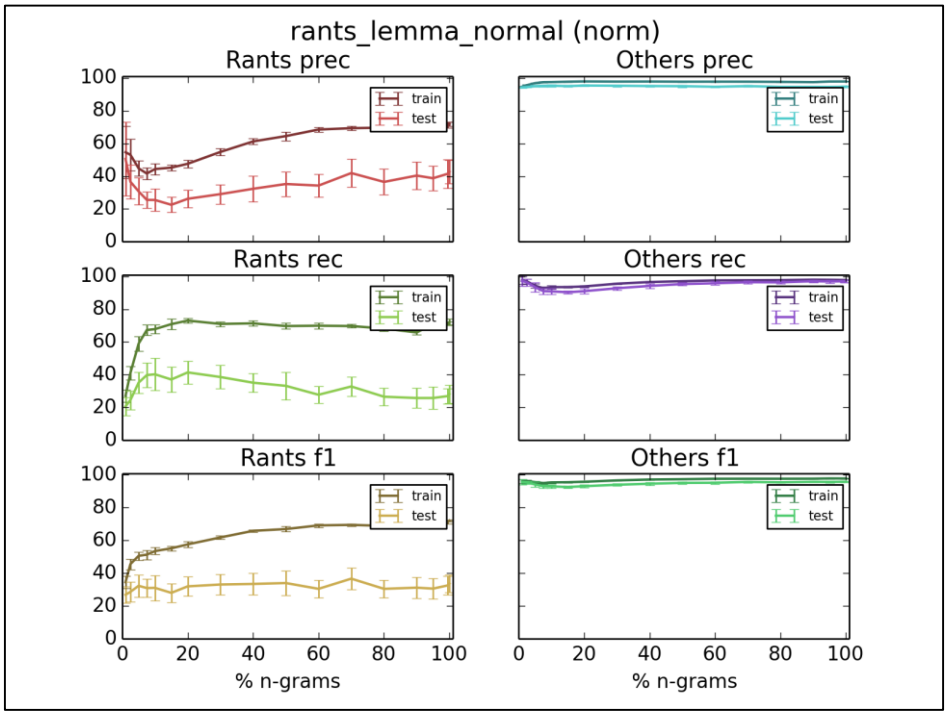

Figure 4: Recall (rec), precision (prec), and f1 measure for classifying Symantec user rants versus non-rants as a function of the percentage of n-grams in the training documents

One possible explanation is the small percentage of rants found in the corpus. To test for this influence, we ran an experiment in which rants were oversampled: The training corpus was repeatedly filled up with rants until rants and non-rants had approximately the same frequency. The results for this experiment can be seen in Figure 5. The f1 performance is similar; however, there is a notable difference for higher n-gram vocabulary percentages: The recall is consistently higher at the cost of lower precision.

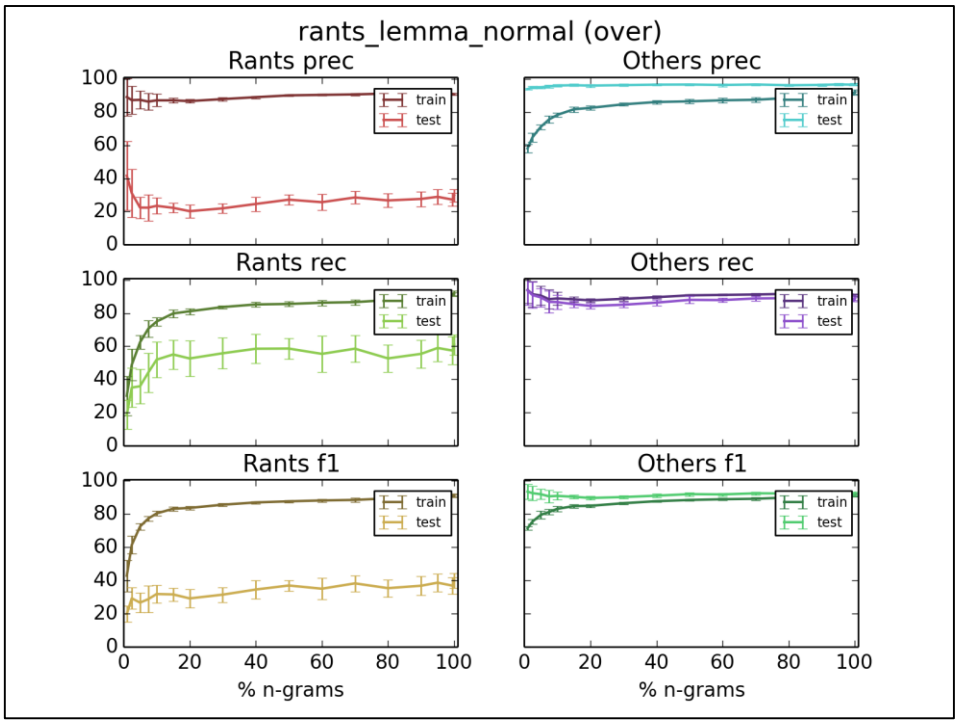

Figure 5: Classification of Symantec user rants with oversampling

1.5.3 Augmenting Symantec Posts Classification with Amazon Reviews

Under the assumption that 1-star Amazon reviews might convey sentiments similar to that of rants, we devised an experiment in which we added 989 Symantec-related Amazon 1-star reviews to the training corpus for rant classification. The n-gram vocabulary was restricted to n-grams occurring in the posts. The results can be seen in Figure 6. Unfortunately, adding the reviews makes performance for rant detection consistently worse. Oversampling helps lessen this effect, but the performance stays below that without Amazon reviews.

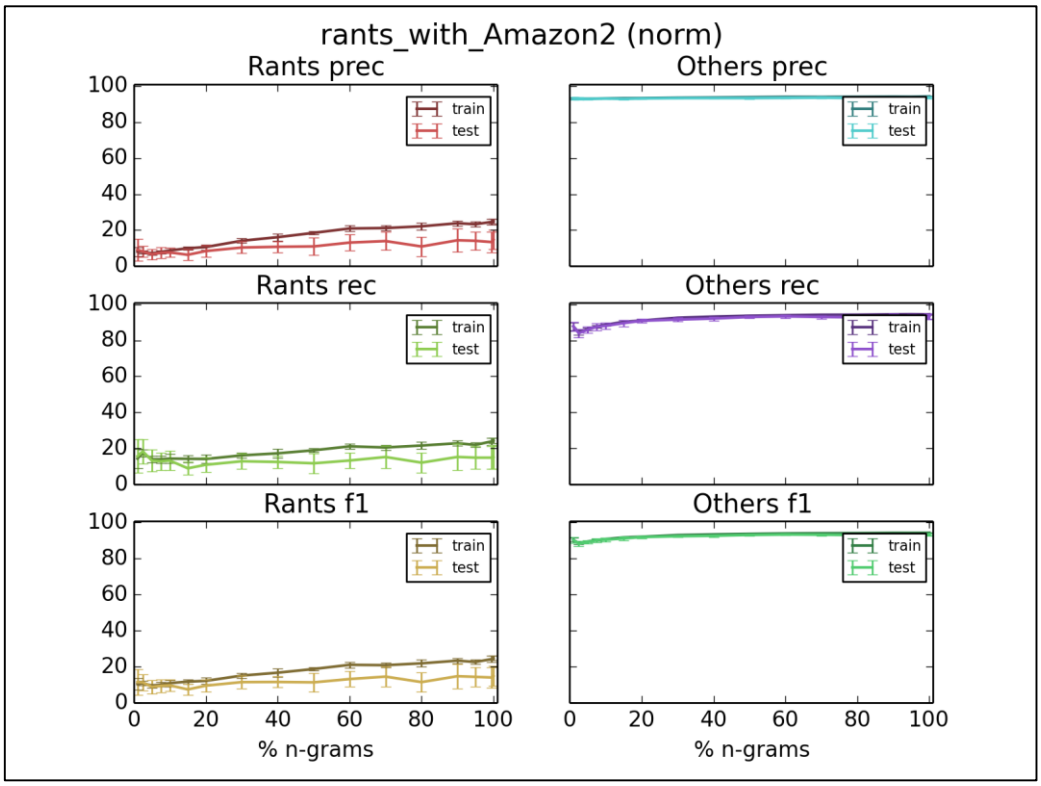

Figure 6: Results for rant classification when augmenting the training corpus with 1-star Amazon reviews for Symantec products

1.5.4 Summary

We set up and evaluated a machine learning pipeline which allows us to filter for n-grams with maximum classification power for sentiment analysis. Unfortunately, the overall performance in the target application of rant detection is comparatively low. Since the setup works well for a corpus of Amazon reviews, the problem might lie with characteristics of the posts corpus, such as its size and the comparatively short posts; we would probably get better results with more posts. Furthermore, since sentiment labelling can also be difficult for humans, the post-classification process itself may have introduced errors into the labels. To check for such errors, we would have to double-check the classification of selected posts, possibly aided by confusion items from the experiments.

At the same time, the results obtained on the Amazon reviews corpus are promising, with an f1 score of consistently above 0.8. To our knowledge, this is the first lexical sentiment classification result obtained on the Amazon review corpus. It compares favourably to similar results based on more complex features from Hu and Liu (2004). Since this result has been obtained by using lexical features derived from domain-specific documents, and the classification algorithm could draw upon a high-dimensional model of these features, we expect this to be a general upper bound for sentiment analysis using Acrolinx rules.

1.6 Sentiment Analysis with Generic Sentiment Lexica

1.6.1 Rules for English Sentiment Analysis

For English, we have derived Acrolinx rules similar to those introduced in Section 1.2, but based on words annotated with sentiment values derived from the SentiWord corpus (Guerini et al., 2013). Words in the corpus are marked with a "prior" sentiment value between -1 (extremely negative) and +1 (extremely positive). For broad applicability, we used approx. two thirds of the range, i.e.,

words with absolute scores above 0.38. In order to find conjugated word forms, we generated Acrolinx trigger patterns from the words' lemmata. We generalized the rules' scopes further by adding simple negation triggers which mark the respective opposite sentiment of the lexical markers. Figure 7 shows parts of the sentiment rules with the negation patterns.

```
#VERSION_2

#ERROR senti_pos

#RULEID EN20140718_173009GEN

#HELP "<html>NA</html>"

#OBJS

@pos_lemma   ::=    [ MORPH.LEMMA "^(ability|able|absolution|abundance|abundant|academic|acceptable|a

@neg_lemma   ::=    [ MORPH.LEMMA "^(2-dimensional|abandon|abandonment|abase|abduct|abduction|abetali

@negation    ::=    [  TOK "^(no|not)$" ];

#RULES

TRIGGER(80) == [-@negation]{4,} @pos_lemma^1
               -> ($keyword)
               -> { mark : $keyword; }

TRIGGER(80) == @negation^1[-@negation]{,3} @neg_lemma^2
               -> ($neg, $keyword)
               -> { mark : $neg, $keyword; }
```

Figure 7: Part of the SentiWords-based lexical rules for sentiment detection

1.6.2 Evaluation of Rules

In order to test the validity of the rules, we used them to classify English 1-star and 5-star Amazon reviews. We conducted machine learning experiments in ten-fold cross-validation: In each experiment, nine of ten partitions (i.e., 90% of the data) were used for training, and the remaining tenth partition (10%) as test data, such that every partition was used as a test partition once.

Training consisted of checking the reviews using the sentiment rules. The resulting numbers of positive and negative flags, normalized by the number of words in the reviews, were used as features for training a linear-kernel SVM to discriminate 1-star from 5-star reviews. Note that this meant the classifier only had two features at its disposal. Table 6 shows the results:

Review	Precision	Recall
1-star	77.36% +/- 2.55%	66.91% +/- 2.91%
5-star	70.27% +/- 2.25%	79.99% +/- 1.83%

Table 6: Classification results for English Amazon reviews with SentiWords-based Acrolinx rules

Precision and recall were almost consistently above 65% and reaching 80% in some cases. This shows that the distribution of positive and negative words alone can be a sufficient indicator of the sentiment polarity of a text.

1.6.2.1 Transfer of the Classification Result

The classification with a linear-kernel SVM lends itself to easy interpretation in different application contexts, since the decision boundary is expressed as a linear equation in terms of the features used for classification, i.e., the flag counts for the positive and negative rules, respectively. For example,

when training on the whole set of Amazon reviews, the sentiment score *s(t)* of a text *t* can be expressed by the following expression, where $|w_t|$ is the number of words in the regarded text, $|p_t|$ is the number of positive rule flags in the text, and $|n_t|$ is the number of negative flags:

$$s(t) = \frac{13.22\ |\mathrm{p_t}| - 33.3\ |\mathrm{n_t}|}{|\mathrm{w_t}|} - 0.29$$

If this sentiment score is zero or above, the sentiment of the text is classified as positive; below zero, the sentiment is classified as negative.

1.6.3 Rules for French Sentiment Analysis

For French sentiment analysis, we drew upon a sentiment-annotated lexicon published in the context of the OpeNER project[6]. The OpeNER lexicon annotates words with positive, neutral and negative sentiment content. Some items are ambiguous in that they can convey more than one sentiment. We used only items which had a unique sentiment annotation in the lexicon to build Acrolinx rules. As with English rules, we used lemma patterns to find and mark inflected forms; we also added simple negation patterns using elements from the OpeNER lexicon annotated as "polarityShifters".

Experiments were conducted in the same manner as described above for the English Acrolinx rules built from the SentiWords corpus. The experiments were run on a subset of 4000 1-star and 4000 5-star small[7] reviews from the French Amazon corpus. Table 7 shows the results for the 10-fold cross-validation experiment set run in this manner. Although weaker than the results for the English rules, the results are still consistently high with an f1 score of approximately .61 for 1-star reviews and .67 for 5-star reviews.

Review	Precision	Recall
1-star	66.70% +/- 2.77%	58.05% +/- 2.73%
5-star	62.87% +/- 2.15%	71.00% +/- 2.70%

Table 7: Classification results for French Amazon reviews with OpeNER-based Acrolinx rules

As with the English rules, we can train a classifier on the whole set of reviews and obtain a linear function for a sentiment score *s(t)* in terms of the number of words $|w_t|$, the number of positive rule flags $|p_t|$, and the number of negative flags $|n_t|$:

$$s(t) = \frac{8\ |\mathrm{p_t}| - 29.4\ |\mathrm{n_t}|}{|\mathrm{w_t}|} + 1.23$$

If for the negative and positive flag counts of a given document this score is below zero, the document's sentiment is classified as negative; otherwise positive.

1.7 Conclusion

We have examined two different approaches for the generation of lexical sentiment features as building blocks for Acrolinx rules: Generating vocabulary from domain-specific corpora and using general sentiment lexica.

Classification results for vocabulary learnt from application domain-specific corpora varied. While discrimination of very negative versus very positive Amazon reviews worked at a remarkable reliability

[6] See http://www.opener-project.eu (Accessed: November 2014).
[7] Between 1000 and 2000 characters.

of more than 0.8 (f1 score), discrimination of rants in user posts can be considered unreliable. Acrolinx rules generated from English and French sentiment lexica worked comparably well for the classification of 1-star and 5-star Amazon reviews. Both rule sets have been integrated into the linguistic resources on the hosted Acrolinx server for ACCEPT. They can be selected as "Sentiment-SentiWords" (for English) and "Sentiment-OpeNER" (for French). For further integrations, the flags generated by the respective rules can be used with a simple decision mechanism based on a linear decision criterion.

2 Report of Evaluation Results

2.1 Objective and Structure

This part of the deliverable describes the results of Task 3.4. The main goal of this task is to analyze the relationship between the classification data obtained in previous tasks of this work package and post-editing data obtained in WPs 7 and 8 (productivity data, quality data, usage data).

The Description of Work mentions two specific objectives:

- to establish whether certain content types lend themselves better to a specific type of post-editing (monolingual or bilingual);
- to investigate whether Sentiment Analysis data can be used to check that translations (after editing) correctly conveys source sentiments.

During the course of the task, we found that both tasks could not be carried out for reasons explained below. We therefore pursued the main goal of Task 3.4 in two other experiments:

- We established whether flags from sentiment rules, as well as pre-editing rules and post-editing rules, can be used to improve the predictions of a machine-translation quality estimator;
- We investigated whether the topics identified for a document correlate with the machine translation output quality determined by an automatic metric.

In the following, we briefly describe the work on the original objectives (Sections 2.2 and 2.3), followed by the main work on the actual experiments (Sections 2.4 and 2.5).

2.2 Original Objective: Using Topics to Select Post-Editing Type

Previously in the project, it has been shown that bilingual post-editing is generally preferred to monolingual post-editing (see Deliverable D7.1.1: *Data and report from user studies – Year 1*). At the same time, presenting the source text gives no advantage when the post-editor has no working knowledge of the source language. Monolingual post-editing, on the other hand, has the implicit advantage of a broader range of potential post-editors. It is thus appealing to identify the documents suitable for monolingual post-editing by means of topic analysis.

This objective could not be investigate directly, since the monolingual and bilingual experiments in WPs 7 and 8 did not collect PE data on the same documents, making a comparison difficult. More profoundly, the topic analyses developed in Tasks 3.1 and 3.2 rely on whole documents. The conducted post-editing experiments, however, had the goal of identifying specific sentence-level issues in the MT output. For example, precise sentence-level post-editing and timing information was collected, and many sentences were post-edited several times by different post-editors. This proved to be very useful

for the development of post-editing rules in Task 2.2. However, this in-depth approach did not produce enough document-level data for a correlation analysis against the classified topics.

2.3 Original Objective: Investigate Whether Sentiment is Correctly Conveyed

Another of the original objectives was to identify whether sentiment is correctly conveyed during machine translation, given that sentiment is particularly common in user-generated content, but might pose a special challenge for MT systems. For this purpose, we planned to use the sentiment classifier from Task 3.3 to test whether the source document and the MT output have the same polarity (positive/negative) as identified by the classifier.

To test the reliability of the classifiers for this bilingual task, we started on this objective by performing a baseline test. We used a set of 250 pairs of forum posts from the Background IP that Symantec contributed to ACCEPT, each consisting of an English source post and a manually-created reference translation. We then ran the English and French sentiment classifiers developed in Task 3.3 on the English source and French target documents to classify the polarity of both sides. As the target documents have been translated by hand, we assumed they would correctly preserve the sentiment of the source document. We thus expected the classifier to compute similar sentiment scores, or at least the same polarity for both sides.

Figure 8 shows the results of the classifier. Each dot corresponds to a document pair, where the sentiment score of the English source is given along the x-axis, and the score of the French target is given along the y-axis. The overall correlation is 0.49 (Pearson's *r*), making the scores somewhat positively related. The significance of this correlation against the null hypothesis of no linear dependency between the two scores is p=1.348e-16, making the correlation statistically highly significant.

However, of the 250 document pairs, the classified polarity (positive/negative) was only preserved in 142 cases, whereas it switched in the remaining 108 cases. When allowing a certain variation between the score values, we found 158 pairs where the score differed by at most 1.0 (grey area in the figure), whereas the score differed by a wider margin for the remaining 92 pairs. This gives a prediction precision of 63.2%. Given that the examined document pairs can be assumed to display near-perfect sentiment preservation, we found it likely that the approach would be too unreliable to correctly measure sentiment preservation on machine-translated documents.

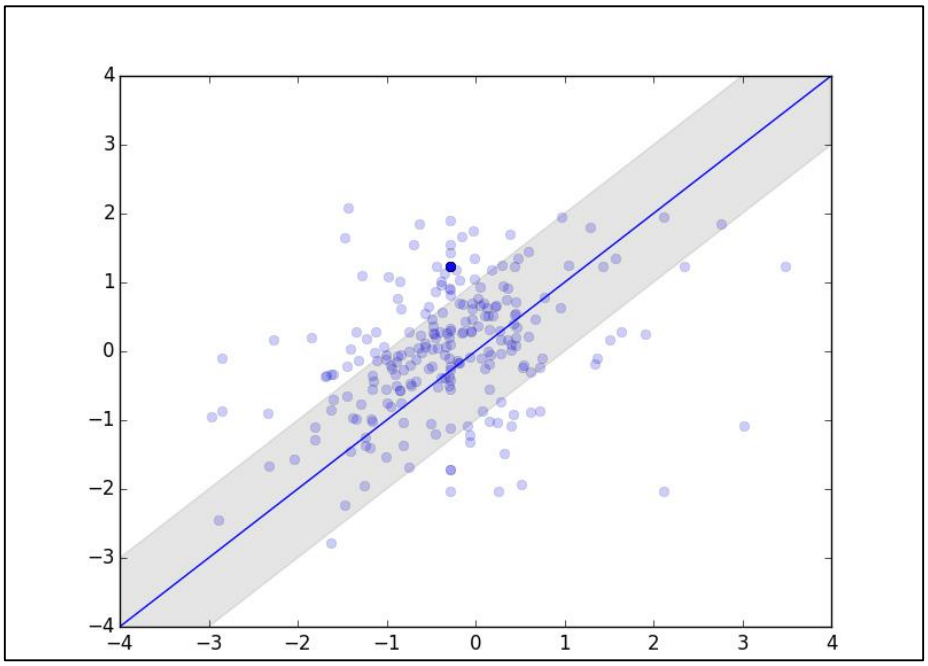

Figure 8: Scatter plot of French sentiment scores as a function of English sentiment scores[8]

2.4 Using Sentiment Classification Data for Sentence-Level Quality Estimation

Instead of correlating the classification data directly with post-editing effort, one possible approach is to rely on machine-translation quality estimation techniques in order to predict specific properties of the machine translation output (e.g., its quality in terms of fluency, adequacy or technical post-editing effort) using features generated by classifiers or resources built in other Tasks of WP3 (e.g., sentiment rules).

This problem can be tackled as a regression task at the sentence level (e.g., Specia et al., 2009). Rather than using sentiment-based features on their own, however, we are interested in finding out whether they can help improve the performance of a baseline machine translation quality estimation (QE) system. Additionally, rather than focusing exclusively on sentiment rules, we also want to investigate the impact of the language checking (pre-editing and post-editing) rules developed in WP2 on quality estimation. The next section describes the methodology and the specific rules that have been considered.

2.4.1 Methodology

The main challenge in relation to the objective described in the previous section concerns the amount of data generated in WPs 7 and 8: it is too small to justify using a quality estimation approach based on machine learning techniques. In the experiments described in this paper, we therefore use the SymForum dataset described in Kaljahi et al. (2014). This dataset consists of 4500 English segments extracted from the English section of the Norton forum and is therefore well-suited for this ACCEPT-related task. These English segments were translated into French using one of the three MT systems:

- The baseline ACCEPT system customised to the Symantec domain (1500 segments);
- A SYSTRAN system customised to the Symantec domain (1500 segments);

[8] The grey area includes documents whose scores differ by at most 1.0.

- Microsoft Translator (1500 segments).

Using two other systems besides the ACCEPT system for which rules were specifically developed in WP2 allows us to check the portability of the rules, since the rules are applied to the entire dataset. Due to the size and nature of the dataset, however, it is not possible to conduct individual experiments for each system. For each translated segment, the dataset includes a post-edited version produced by a professional translator, as well as a set of three human judgments (fluency and adequacy ratings). For the present experiments, we use the post-edited version to generate TER and BLEU scores for each of the 4500 segments.[9] We use the average fluency and adequacy scores assigned to each of the 4500 segments. In these experiments, the 1-5 fluency and adequacy scores are scaled to 0-1 to match the range of the automatic metrics. To build quality estimation systems, we split the set of 4500 segments into three sets: 3000 segments are used for training, 500 for tuning *C* and *gamma* values using grid search in terms of Root Mean Square Error (RMSE), and 1000 for evaluation purposes. The *scikit-learn* machine learning framework (Pedregosa et al., 2011) is used for the experiments, which all rely on the following regression approach: Support Vector Machine epsilon-SVR with RBF kernel. In order to extract the 17 features used to build a baseline system as well as combined systems, we use the QuEst tool (Specia et al., 2013).[10]

The sentiment and language checking features investigated here are extracted in the following manner: each source, English segment is checked in terms of spelling, grammar and style violations as well as sentiment, using the Acrolinx language checking system made available by the ACCEPT framework.[11] For this purpose, we used the pre-editing rules presented in Deliverable D2.2: *Definition of pre-editing rules for English and French* and the sentiment rules presented in the first part of this deliverable. Similarly, each French machine-translated segment is checked using the same system configured with French resources. Additionally, the French output is checked using dedicated post-editing rules, presented in Deliverable D2.4: *Definition of post-editing rules for English, French, German and Japanese*. Table 8 shows the various types of extracted features in a short form; please refer to Table 11 in the appendix for a complete list of the considered rules.

Feature set	Features #	Feature sources
sentiment	4	Flag counts from the 4 sentiment rule set developed in Task 3.3
source	4	Flag counts from the English pre-editing forum rule set, grouped in the categories grammar, style, punctuation and spelling.
target	4	Flag counts from the French pre-editing rule set, grouped in the categories grammar, style, punctuation and spelling.
both	8	Combination of *source* and *target* feature sets
source_total	1	Sum of all *source* flags
target_total	1	Sum of all *target* flags
target2_total	1	Sum of flags from the French post-editing rule sets
both_totals	2	Combination of *source_total* and *target_total* feature sets

Table 8: Short description of feature sets

[9] To calculate BLEU scores, version 13a of the MTEval script is used at the segment level (thus performing smoothing). Since TER scores can be higher than 1 when the number of errors is higher than the segment length, they are cut off at 1 and reversed to be comparable to the other scores.
[10] http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17 (Accessed: December 2014).
[11] The default rule classification provided by the Acrolinx system was slightly modified to create a "punctuation" category for instance.

2.4.2 Results

The results obtained using the various quality estimation systems are presented in Table 9. Each of the feature sets described in Table 8 was combined with the baseline feature set.

System	fluency		adequacy		bleu		ter	
	RMSE	R	RMSE	R	RMSE	R	RMSE	R
baseline (17)	0.2409	0.4937	0.2207	0.4625	0.2785	0.3318	0.2285	0.3327
baseline+sentiment (21)	**0.2427**	**0.4804**	**0.2232**	**0.4411**	0.2794	0.3219	0.2293	0.3181
baseline+source (21)	0.2403	0.4949	0.2209	0.4636	**0.2801**	0.3256	0.2329	0.3022
baseline+target (21)	**0.2368**	**0.5209**	**0.2190**	**0.4807**	**0.2741**	**0.3725**	0.2324	**0.2670**
baseline+both (25)	**0.2372**	**0.5178**	**0.2187**	0.4785	0.2779	0.3371	0.2320	**0.2704**
baseline+source_total (18)	0.2402	0.4957	0.2207	0.4675	0.2784	0.3316	0.2265	0.3436
baseline+target_total (18)	**0.2384**	**0.5111**	0.2197	**0.4752**	**0.2767**	**0.3545**	0.2270	0.3408
baseline+target2_total (18)	**0.2387**	**0.5069**	0.2201	**0.4710**	**0.2766**	**0.3489**	0.2327	0.3267
baseline+both_totals (19)	**0.2383**	**0.5111**	**0.2190**	**0.4786**	**0.2754**	**0.3596**	0.2262	0.3479

Table 9: Quality Estimation Results: RMSE (lower is better) and Pearson R (higher is better)[12]

These results show that the sentiment-based features do not help improve the performance of a (strong) baseline quality estimation system. These results also show that the baseline+target system and baseline+both_totals are the best systems. These systems perform significantly better than the baseline system for three of the metrics (adequacy, fluency and BLEU). When individual target features are combined with the individual source features in the baseline+both system, performance degrades for some of the metrics, which suggests that the individual source features do not contribute as much useful information as the target features. We do not observe this trend when the numbers of total errors are used in the baseline+both_totals system, indicating that some source features are more informative than others. Another important observation is that the baseline+target system is outperformed by the baseline system for the prediction of TER scores. This suggests that individual target features are causing some degradation, which would warrant further investigation. Finally it is worth pointing out that the baseline+tgt2_total performs in a manner consistent with the baseline+target_total, which indicates that the rules developed for checking content in a pre-editing scenario are as effective as rules developed for checking content in a post-editing scenario (at least, as far as quality estimation is concerned).

2.5 Correlating Topic Classification Data with Document-Level Quality Metrics

To estimate machine translation quality at the document level, we used a set of 157 English documents from the NGOs *AMREF* and *Action contre la faim* that had been manually translated from English to French and which had not been used to train the MT system. We classified the English source documents along 10 topics using the classifier developed in Task 3.2. This resulted in a list of 10 topic similarity features for each document. Please refer to Deliverable D3.2: *Taxonomy of NGO content and rules for automatic classification* for details on the topic classification and the similarity measure.

[12] Systems performing significantly better than the baseline systems are highlighted in green, while systems performing significantly worse than the baseline systems are highlighted in red. To check whether differences between systems are statistically significant, bootstrap resampling is used (Koehn, 2004).

We then translated the documents using the TWB baseline MT system developed in WP4, and computed the BLEU metric for each of the French output documents with respect to their French reference translations.

Finally, we computed the correlations (Pearson's *r*) between the BLEU scores and each of the 10 topic similarity value across all documents. If we found a positive correlation between the similarity of the source documents to a specific topic on the one hand, and the similarity of the corresponding target documents to the reference translation as measured by BLEU on the other hand, then this would imply that documents pertaining to the respective topic lend themselves to easier post-editing.

Topic	Most relevant words					Pearson's r	p-value
#1	food	project	product	local	market	-0.0572	0.3776
#2	program	water	staff	area	project	-0.0018	0.978
#3	health	water	train	assess	point	0.0958	0.139
#4	child	health	care	drug	mother	0.042	0.5177
#5	case	patient	treatment	infect	day	-0.1249	0.0534
#6	child	project	train	report	data	-0.096	0.1379
#7	child	health	school	research	inform	0.0665	0.3051
#8	stock	request	good	order	click	0.0509	0.4326
#9	say	see	staff	transport	good	-0.0092	0.8868
#10	week	contract	infect	fever	year	0.0388	0.5492

Table 10: Identified topics and their correlation to BLEU scores

Table 10 shows the results. Unfortunately, we found little to no correlation between these two measures (Pearson's *r*), and the results were mostly not significant ($p>0.05$). This was also the case if we removed all documents for a given topic that had no similarity to that topic. The results could possibly be improved to a certain extent by using a smarter automatic metric, a *document-level* quality estimation technique or even manual translation judgments. However, given the very low starting point, we are not confident that this would lead to significant improvements in the correlation.

2.6 Conclusion

We have shown that the sentiment classifier cannot reliably be used to determine whether sentiment is preserved during the MT translation. Also, we used the sentiment flags as features for quality estimation, but found that they do not help to improve the quality estimates. However, we found that the post-editing rules developed in Task 2.2 provide good indicators for the quality of the MT output and improved the baseline system.

As for the topic classification, we conclude that the topics as identified by our classifier are not useful indicators for the quality of the machine translation of NGO documents using the given system, and thus likely do not correlate with the required post-editing effort. We stipulate that topic classification could indeed become relevant when considering documents that are entirely outside the training domain of the used SMT system. This use case, however, is beyond the scope of the ACCEPT project.

References

- Marco Guerini, Lorenzo Gatti, Marco Turchi:
  Sentiment analysis: How to derive prior polarities from SentiWordNet.
  In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, pages 1259–1269, Seattle, Washington, USA, 2013.

- Minqing Hu, Bing Liu:
  Mining and summarizing customer reviews.
  In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, Washington, USA, 2004.

- Nitin Jindal, Bing Liu:
  Opinion spam and analysis.
  In: *Proceedings of 1st ACM International Conference on Web Search and Data* Mining (WSDM), pages 219–230, Standford, USA, 2008.

- Rasoul Kaljahi, Jennifer Foster, Johann Roturier:
  Syntax and semantics in quality estimation of machine translation.
  In: *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, pages 67–77, Doha, Qatar, 2014.

- Philipp Koehn:
  Statistical significance tests for machine translation evaluation.
  In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 4, pages 388–395, Barcelona, Spain, 2004.

- Bing Liu:
  Sentiment Analysis and Opinion Mining.
  Morgan & Claypool Publishers, 2012.

- Andrew McCallum, Kamal Nigam:
  A Comparison of event models for Naïve Bayes text classification.
  In: *Proceedings of Workshop on Learning for Text Categorization (AAAI-98)*, volume 752, pages 41–48, Madison, Wisconsin, USA, 1998.

- Bo Pang, Lillian Lee, Shivakumar Vaithyanathan:
  Thumbs up? Sentiment classification using machine learning techniques.
  In: *Proceedings of the ACL conference on Empirical methods in natural language processing (EMNLP)*, volume 10, pages 79–86, Philadelphia, USA, 2002.

- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay:
  Scikit-learn: Machine learning in Python.
  Journal of Machine Learning Research, 12: 2825–2830, 2011.

- Lucia Specia, Kashif Shah, Jose GC De Souza, Trevor Cohn:
  Quest-a translation quality estimation framework.
  In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sophia, Bulgaria, 2013.

- Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, Nello Cristianini:
  Estimating the sentence-level quality of machine translation systems.
  In: *Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT-2009),* pages 28–35, Barcelona, Spain, 2009.

- Sida Wang, Christopher Manning:
  Baselines and bigrams: Simple, good sentiment and topic classification.
  In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 90–94, Jeju Island, Korea, 2012.

Appendix 1

Feature set	Features #	Feature sources
sentiment	4	Source: “senti_words_neg” and “senti_words_pos” flags from *Sentiment-SentiWords* rule set Target: “opeNER_neg” and “opeNER_post” flags from *Sentiment-OpeNER* rule set
source	4	Flags from English *Preediting_Forum* rule set, grouped in the following 4 categories: grammar = ["wrong_comparative_or_superlative", "missing_word", "np_number_agreement", "to_too_confusion", "fewer_less_confusion", "wrong_word", "noun_adjective_verb_confusion", "its_it_is_confusion", "wrong_verb_form", "subject_verb_agreement", "avoid_duplicates", "than_then_confusion", "there_their_confusion", "write_words_together", "loose_lose_confusion", "irregular_verb_use", "a_an_distinction", "repeated_word", "use_verb_with_object_and_infinitive", "where_were_confusion", "use_verb_with_to_and_infinitive", "much_many_confusion", "wrong_sequence_of_words", "repetition_flag", "use_preposition_with_ing_verb", "incorrect_preposition", "uncountable_nouns", "use_verb_with_object_and_to", "noun_adjective_confusion"] style = ["avoid_colloquialism_and_metaphorical_language", "sentence_too_long", "do_not_use_this_word", "use_will", "do_not_use_whether_or_not"] punctuation = ["missing_space", "use_comma_after_subordinate_phrase", "use_comma_with_parenthetical_expressions", "use_comma_after_introductory_phrase", "duplicate_punctuation_mark", "no_space_between_number_and_word", "use_end_of_sentence_punctuation", "incorrect_extra_comma"] spelling = ["spelling_flag", "spelling_error", "unknown_word", "must_hyphenate"]
target	4	Flags from French *Preediting_Forum* rule set, grouped in the following 4 categories: grammar = [“abreviationIncorrecte”, “accord_phrase_nominale”, “accord_sujet_verbe”, “confusion_futur_conditionnel”, “confusionParticipe”, “elidez_ce_mot”, “evitez_adverbes”, “evitezAlors”, “evitez_ce_qui”, “evitez_conditionnel”, “evitez_le_participe_present”, “evitez_les_phrases_clivees”, “evitez_lettres_entre_parentheses”, “evitez_une_conjonction_en_debut_de_phrase”, “expression_incorrecte”, “forme_verbale_incorrecte”, “mettez_imperatif”, “negation_incomplete”, “negationIncorrecte”, “ne_pas_elider”, “passe_compose_avec_etre”, “repetezSujet”, “repetition_flag”, “sentence_skipped”, “sequence_incorrecte_de_mots”, “utilisez_de”, “utilisez_des”] punctuation = [“ajouterTiret”, “ajouterVirguleApresPP”, “ajoutez_un_blanc”, “ajoutez_une_virgule”, “erreur_de_majuscule”, “espace_en_trop”, “espaces_autour_ponctuation”, “evitez_ponctuation”, “evitez_toute_une_phrase_en_majuscule”, “ponctuation_double”, “ponctuation_incorrecte”, “tiret_sans_espace”, “utilisez_une_majuscule_en_debut_de_phrase”] style = [“evitez_le_langage_familier”, “evitez_les_anglicismes”, “evitez_les_questions_directes”, “sentence_too_long”, “evitezAbrevForum”, “evitez_est_ce_que”]

		spelling = [“homophonesDivers”, “homophones_verbe_nom”, “la_vs_la”, “mot_inconnu”, “mot_inconnu_en_capitales”, “on_vs_ont_vs_sont”, “ou_vs_ou”, “spelling_flag”, “sur_vs_sur”, “tous_vs_tout”, “a_vs_a”, “ce_vs_se”, “des_vs_des”, “du_vs_du”]
both	8	Combination of *source* and *target* feature sets
source_total	1	Sum of all *source* flags
target_total	1	Sum of all *target* flags
target2_total	1	Sum of flags from the following rule sets: Set1_PostEd_Sym, Set2_PostEd_Sym, Set3_PostEd_Sym, Set4_PostEd_Sym, Set5_PostEd_Sym, Set6_PostEd_Sym, Set7_PostEd_Sym, Set8_PostEd_Sym
both_totals	2	Combination of *source_total* and *target_total* feature sets

Table 11: Complete description of feature sets