Visualizing Sentiment Analysis on a User Forum

Rasmus Sundberg[†] **Anders Eriksson**[†] **Johan Bini**[*] **Pierre Nugues**[†]

[†] Lund University
Department of Computer science
Lund, Sweden
rasmus.e.sundberg@gmail.com, dt06ae2@gmail.com,

[*] Qliktech
Scheelevägen 24-26
Lund, Sweden
johan.bini@qliktech.com, Pierre.Nugues@cs.lth.se

Abstract

Sentiment analysis, or opinion mining, is the process of extracting sentiment from documents or sentences, where the expressed sentiment is typically categorized as positive, negative, or neutral. Many different techniques have been proposed. In this paper, we report the reimplementation of nine algorithms and their evaluation across four corpora to assess the sentiment at the sentence level. We extracted the named entities from each sentence and we associated them with the sentence sentiment. We built a graphical module based on the Qlikview software suite to visualize the sentiments attached to named entities mentioned in Internet forums and follow opinion changes over time.

Keywords: sentiment analysis, classification algorithms, visualization

1. Sentiment Analysis

An ever increasing number of web sites – user forums, newspapers, merchant sites –, have feedback features, where users can leave comments, opinions, or textual evaluations on products, people, or facts. Such opinions have an increasing influence in our decisions. Pang and Lee (2008) describe that 60% of US residents have done online research on a product at least once. Collecting snapshots of such opinions is now considered crucial to many analysts.

Sentiment analysis or opinion mining is the process of extracting sentiment from fragments of text. The granularity level is usually a sentence or a document and the expressed sentiment is typically categorized as *positive*, *negative*, or *neutral*.

This paper reports the implementation nine algorithms based on machine–learning techniques and their evaluation across four corpora. We extracted the named entities from the sentences and we used these algorithms to visualize the sentiment associated with specific entities, typically product names, and their evolution in time.

2. Corpora

We used four corpora containing data with a sentiment annotation at the sentence level. They consist of:

- The Multiple-Perspective Question Answering corpus (MPQA) containing text from newspapers annotated at the expression level (Wiebe et al., 2005). The expressions were used as sentences.
- The SICS corpus created by Täckström and McDonald (2011) containing reviews from different domains. This corpus had a fourth annotation tag corresponding to *off-topic*; we converted it to neutral.
- The English part of a corpus of book reviews gathered by Zagibalov et al. (2010). This corpus is referred to as EnBooks and was originally annotated at the document level. We split it into sentences and we manually annotated all the sentences.
- A small set of sentences gathered from Qlikview's forum[1] that we manually annotated. Qlikview forums deal with various aspects of Qlikview products and software.

3. Algorithms

We implemented nine algorithms to classify the sentiment expressed in each sentence of our corpora. We used the three standard categories: *positive*, *neutral*, and *negative*. We summarize the algorithms here:

MajVote uses the basic majority voting algorithm. The simplest way of classifying a sentence with polarity is to count the number of positive and negative words it contains. This approach is brittle for various reasons. The major problem is the diversity of expressions that can occur. Another problem is that there is no guarantee that the word polarity corresponds exactly to the sentence polarity.

VoteFlip is a voting method that deals better with negations. It reverses the polarity of a sentence if there is an odd number of negating words in a sentence (Choi and Cardie, 2009). The issue with vote flip is that it cannot deal with the large variety of negation expressions.

Bayes is the naïve Bayes method trained on a bag-of-words model; see the description in Sect. 6.. We used the naïve Bayes implementation from the Weka collection (Hall et al., 2009).

SVM are support vector machines trained on a bag-of-words model. The input is the same as for naïve Bayes. We used a grid search from the LIBSVM library (Chang and Lin, 2011) to optimize the parameters and find the best type of SVM and kernel configuration. The result was C-SVC with a linear kernel.

[1] `http://community.qlikview.com/`

LogReg is a logistic regression using the LIBLINEAR implementation (Fan et al., 2008). As for SVMs, the input is the same as for naïve Bayes. We evaluated the available solver types and we achieved the best performance with the dual L2-regularized solver.

HCRF is the hidden conditional random fields method described in Nakagawa et al. (2010), similar to the method by Täckström and McDonald (2011). This method uses features in a sequence model instead of bags of words. We used the HCRF library by Wang et al. (2006). Unfortunately, we could not work out the sparse representation of data which is necessary for a large dataset with thousands of features. We worked it around by excluding words from the dataset.

HCRF+ corresponds to the same method as above but with features from semantic role labeling, grammatical relations, the parse tree, and a polarity reversal dictionary.

AB is the AdaBoosted versions of the previously listed methods. AdaBoosting was introduced by Freund and Schapire (1995). HCRF and voting have not been included in any ensemble methods.

Voting, Stacking, and Bagging were all constructed with naïve Bayes, SVM, and logistic regression as base classifiers. The voting and stacking were done with one of each classifier. Bagging was done with 10 logistic regression classifiers. The stacking version used logistic regression as final classifier.

4. System Architecture

Figure 1 shows the main components of our Automatic Sentiment Analysis (ASA) program. The square boxes correspond to one or more CPU threads in ASA. Note that the sentence polarity corpus is only used at training time.

The visualization component of the sentiment analyzed by ASA is a completely separate entity. Data exchange is handled through files or databases.

In the HCRF+ algorithms, we used syntactic and semantic features in addition to the word sequences. To carry out parsing, ASA uses three different syntactic parsers: OpenNLP, StanfordNLP by Klein and Manning (2003), and the LTH parser by Björkelund et al. (2009). All parsers can detect sentences, lemmas, disambiguate parts of speech, and generate a parse tree.

StanfordNLP and OpenNLP feature a named entity recognition module, while the LTH parser can carry out a semantic role labeling.

5. Automatic Construction of Dictionaries

We constructed a polarity reversal word dictionary. We generated it from the *General Inquirer* database[2] using the words in the categories *negate* and *decrease*.

Because of the cost associated with the manual annotation of datasets, we implemented methods for the automatic construction of word dictionaries. The purpose was to generate a word polarity corpus. The basis is a Markov random walk in a graph of words (a thesaurus). The links can be synonyms or other relations. The method uses a small seed of polar words marked as positive or negative. A random walk is started from every word in the dictionary. If it hits a polarity word, the negative or positive polarity of the starting word is increased based on the length of the walk. Constraints are added to speed up the process and prevent infinite loops.

The main reason that a random walk is used instead of deterministic methods is that the distances between words of opposite polarity can be short. As an example, *good* and *bad* are closely related to each other in WordNet by the 5-word synonym sequence *good, sound, heavy, big,* and *bad*.

The method was first developed by Hassan and Radev (2010). WordNet (Miller, 1995) was used as thesaurus. It was further extended using the *People's Dictionary of Synonyms*[3] for Swedish. In the study, Rosell and Kann (2010) took advantage of the strength of relations available in the dictionary.

We implemented the basic method by Hassan and Radev (2010). However, because of the relative small amount of relations in WordNet, the resulting set of 970 items was deemed too small. We used a manually-created dataset instead based upon the *Subjectivity Lexicon* (Wilson et al., 2005). The algorithm could still be useful for customizing classifiers to new domains by expanding domain-specific words.

6. Evaluation and Experiments

6.1. Experimental Setup

We used all the corpora mentioned in Sect. 2. We extracted 999 sentences from the data set to build the test set and we used the rest as a training set. The test set sentences were taken evenly from all the classes.

The features used in the bag-of-words setting are: a lemma vector where we indicate if a lemma is present or not in the sentence, the part-of-speech counts for all the parts-of-speech values occurring in the sentence, the counts of positive, negative, and neutral words, the sentence length, and the count of negating words. We ran the HCRF method in two forms: using the word sequence and augmented with features extracted from the parse trees and semantic dependencies. These methods are denoted respectively HCRF and HCRF+ in Table 1.

6.2. Overall Results

Table 1 shows the results we obtained for each classification method. We also implemented some ensemble methods and Table 1b shows their performance. We used the standard metrics: precision (P), recall (R), and harmonic mean F_1. We used the mean of F_1 to determine which method had the best results. The samples were taken evenly from all the classes. In Table 1a, bold text and asterisk means a statistical significance at a 95% level.

[2] http://www.wjh.harvard.edu/~inquirer/

[3] http://folkets-lexikon.csc.kth.se/folkets/folkets.en.html

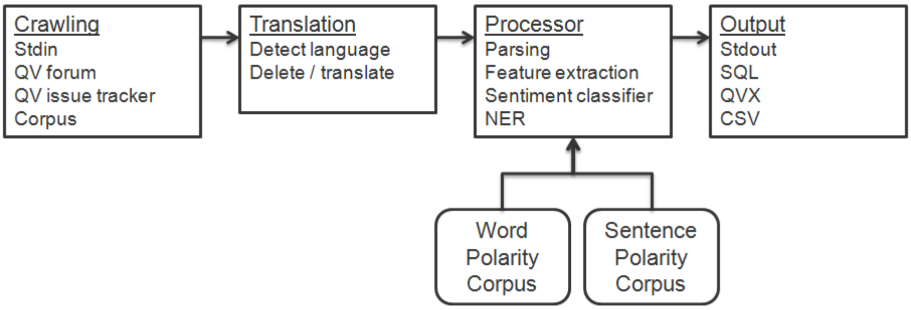


Figure 1: Pipeline diagram

	Neu			Pos			Neg			Mean
Method	P	R	F_1	P	R	F_1	P	R	F_1	$\overline{F}_1$
MajVote	.409	.464	.435	.459	**.696***	**.553***	.581	.204	.302	.430
VoteFlip	.409	.464	.435	.498	**.642***	**.561***	.500	.288	.466	.454
Bayes	.397	.162	.230	.467	**.636***	**.538***	.477	**.589***	.527	.431
SVM	**.454***	**.545***	**.495***	**.695***	.494	**.577***	.530	**.577***	.553	**.542***
LogReg	**.484***	**.506***	**.495***	**.670***	.539	**.598***	.530	**.610***	.567	**.553***
HCRF	.265	.393	.320	.443	.502	.470	**.682***	.476	.560	.450
HCRF+	.401	.487	.440	.547	.511	.528	**.646***	**.584***	**.613**	**.527***

(a) Single classifier versions

ABBayes	.447	.377	.409	.592	.599	.596	.509	.583	.543	.516
ABSVM	.464	.50	.481	.650	.497	.563	.530	.613	.568	.537
ABLogR	.472	.446	.458	.616	.536	.573	.520	.616	.564	.532
Voting	.478	.488	.483	.659	.542	.595	.530	.613	.568	.549
Stacking	.481	.500	.409	.667	.542	.598	.526	.604	.562	.523
Bagging	.469	.503	.486	.662	.524	.585	.529	.601	.563	.545

(b) Ensemble versions

Table 1: Evaluation of the implemented methods

SVM and logistic regression obtained the best results. However, because our corpora were from different domains, our overall results are lower than what we could expect if the corpus was from one domain only. None of the ensemble methods was significantly different in performance from SVM or logistic regression and we set them aside. As an interesting note, the performance of AdaBoost on generative models like naïve Bayes is generally reported as poor, but here we obtained results that showed a large performance increase. For more information on boosting naïve Bayes, see Kim and Kim (2004).

We carried out the rest of the experiments with HCRF, SVM, and logistic regression because of their superior performance. We did not use the SRL features for the HCRF method, then.

6.3. Learning Curve

Figure 2 shows the learning curve for SVM and logistic regression. We used the corpora described in Sect. 2. and the test set from Sect 6.1. and we computed the mean F_1 values as a function of the training set size. The performance increases steeply with the size of the training set up to 3,000 sentences. Beyond 4,000, the slope is much flatter.

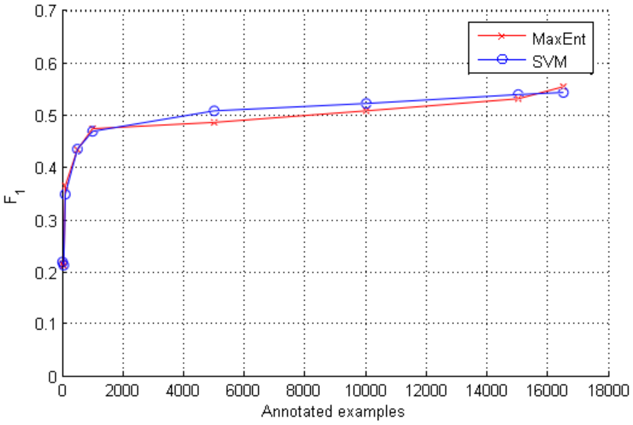


Figure 2: The learning curve.

6.4. Impact of the Training Corpus

We tried to evaluate the impact of the training corpus, when the data sets used for training and testing the algorithms were from different domains.

We first computed a baseline using in-domain training and testing data and a 10-fold cross-validation (CV). The CV accuracy is calculated by dividing the number of correct classifications by the total number of classifications. Table 2a shows the results for SVM and logistic regression as CV is not implemented in the HCRF library. For the SICS corpus, we present the results per product category: books, DVDs, electronics, games, and music. As expected with homogeneous domains, the figures are relatively high, especially when the corpus size is sufficiently large.

To compute the corpus impact, we used the EnBook data set as training set, because it is sufficiently large with 5589 sentences and we tested the resulting model on the other corpora. Table 2b shows the results we obtained. As for the baseline, we present the results for the SICS corpus per category.

Without surprise, the results are lower than those we obtained with the in-domain training data and cross-validation. The figure obtained with the *SICS: Books* subcorpus is the lowest of all. This is somehow paradoxical as the domain – books – seems to be closer to that of EnBooks than to the other subcorpora.

Dataset	**SVM**	**LogR**	**Samples**
SICS: Books	52.8%	66.8%	781
SICS: DVDs	48.1%	48.2%	857
SICS: Electronics	55.9%	55.4%	667
SICS: Games	54.5%	54.5%	1142
SICS: Music	51.2%	51.1%	677
Qlikview	63.0%	60.6%	165
MPQA	66.3%	66.8%	7677
EnBooks	70.8%	71.2%	5589

(a) Results of a 10× cross-validation with the SVM and logistic regression classifiers. The third column is the corpus size.

Dataset	**HCRF**	**SVM**	**LogR**
SICS: Books	22.2%	42.6%	40.3%
SICS: DVDs	45.5%	45.5%	44.0%
SICS: Electronics	44.9%	45.3%	48.3%
SICS: Games	40.6%	48.6%	47.9%
SICS: Music	43.4%	48.4%	49.8%
Qlikview	51.0%	53.1%	49.8%
MPQA	40.0%	39.9%	42.0%
EnBooks	-	-	-

(b) Results on the corpora with EnBooks used as training set.

Table 2: Impact of the training corpus on accuracy.

7. Associating Sentiments to Entities

Our goal was to track the sentiment regarding entities mentioned in user forums, typically products or people, and make them easily comprehensible to an analyst. The algorithms described in Sect. 3. classify a sentence sentiment and not the sentiments on entities.

To associate sentiments to entities, we applied the following principle: The sentiment attached to an entity mentioned in a sentence is exactly that of the sentence. Although this is an approximation, this allowed us to create a mapping of the sentiments expressed in the sentences onto the named entities.

The named entity recognition in ASA uses the library by Finkel and Manning (2009) with a combination of machine learning and regular expression dictionaries. It detects entities corresponding to common categories like person, location, and organization.

Because of the large amount of categories in the Qlikview forums, we had to use dictionaries. They include Qlikview-specific terms like the client versions, services, and script languages, etc.; see Table 3 for some examples. We identified the corresponding names with a combination of dictionaries and hand-written rules.

Item	**Text**
QV9SR3	Version 9 SR3 was available for me.
QV9SR6	I am using QV V. 9 with SR6.
QV10	The Qlikview version number is 10.00.8811.6.

Table 3: Examples of named entities corresponding to Qlikview releases.

8. Visualization

We developed a visualization dashboard to display the results of the classification analyses. We carried out this development with the posts from the Qlikview forums and we used logistic regression as classification method.

Sentiment analysis constructs tabulated numerical data from raw unstructured data. We found that a combination of text with graphical pictures and colors was the best way to convey the results.

8.1. Colors

Following Few (2006), we used the traffic light metaphor, where green indicates that something is good, red is bad, and yellow is neutral. Such a color code seems widely accepted, although it is not completely legible to the 10% of males and 1% of females who are color-blind.

Some visualizations have mixes of colors to indicate opinions between positive, negative, and neutral. We created the color mix using the equation:

$$\frac{\sum \text{positive} - \sum \text{negative}}{\sum \text{positive} + \sum \text{neutral} + \sum \text{negative}}.$$

This gives a number between -1 to 1, which can be used in a lookup in a gradient for a smooth transition.

8.2. The Dashboard

We created a dashboard to visualize the analyses. It consists of three components:

- The main dashboard gives an overview of the analysis;
- the user can select an item and examine trends using the trend dashboard.
- Finally, a third component can display the text with analysis results at the sentence level.

Figure 3 shows the main dashboard, where the sentiment analyses are broken down by forum. Opinion changes are often very important to the analysts and we grouped the results according to the post date.

The user can change the level of details. In the upper part of Figure 3, s/he can select a specific forum and then a specific entity. Figure 4 shows the results and the sentiment over time in one forum. Figure 4a shows a presentation mode with stacked values that use the traffic light metaphor, while Figure 4b shows averaged color values.

The colors of the visualized items, either sets of forums, single forums, or entities, are dynamically computed from the counts of positive, negative, and neutral sentiments each item contains.

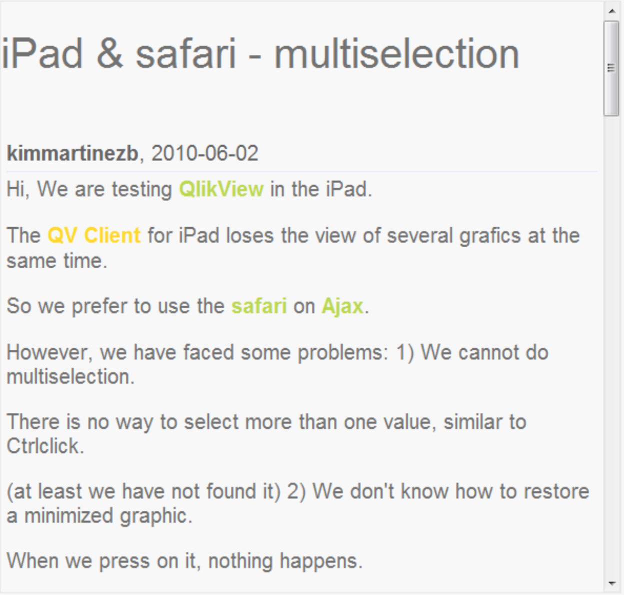
8.3. The Trend Dashboard

Figure 5 shows the dashboard of trends. It reflects the sentiment evolution over time. It follows the same design as the main dashboard but with the relative sentiment changes over time. In the upper left corner, the user can select a date, a time, as well as periods. In the upper right corner, s/he can select one or more forums.

8.4. Visualization of Text

Because sentiment analysis is not totally accurate, it is important to retrieve the results of an analysis in the text and verify if they are correct. We built a component to visualize text with the detected entities and the corresponding sentiment.

Figure 6 shows the results of this visualization. Only named entities are highlighted; the sentiment belongs to the entire sentence though. The sentiment of any sentence can be seen by hovering the mouse pointer over it. The opinion for it will be highlighted in either green, yellow, or red.



iPad & safari - multiselection

kimmartinezb, 2010-06-02

Hi, We are testing QlikView in the iPad.

The QV Client for iPad loses the view of several grafics at the same time.

So we prefer to use the safari on Ajax.

However, we have faced some problems: 1) We cannot do multiselection.

There is no way to select more than one value, similar to Ctrlclick.

(at least we have not found it) 2) We don't know how to restore a minimized graphic.

When we press on it, nothing happens.

Figure 6: Visualization of a forum thread.

9. Conclusion and Future Work

We implemented a set of algorithms to carry out sentiment analysis. Although the performances are slightly behind state-of-the-art implementations, we believe they are competitive. In our study, support vector machines and logistic regression achieved the best performances.

There is room for improvements. We generated learning curves that showed no need for more examples. This means that we probably need more features or a more complex data model to increase the performance.

The route to a more complex data model was chosen with the HCRF method. Because the HCRF implementation could not use words, we cannot draw a definitive conclusion about its performance. The HCRF method was still better than naïve Bayes using much less features. Using features from SRL and the parse tree resulted into a significantly better performance.

We extracted the named entities from each sentence and we associated them with the sentence sentiment. We integrated the sentiment analysis into a visualization tool based on the Qlikview software suite to visualize the sentiment attached to named entities and follow opinion changes over time. We believe such an interface makes the analysis results easier to access and understand.

ASA is programmed in Java and is licensed with GNU General Public License (GPL[4]).

10. References

A. Björkelund, L. Hafdell, and P. Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of The Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 43–48, Boulder, June.

C.C. Chang and C.J. Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

Y. Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of EMNLP*, volume 2, pages 590–598, Singapore.

R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. 2008. LIBLINEAR: A library for large linear classification. *Machine Learning Research*, 9:1871–1874.

S. Few. 2006. *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media, Cambridge, Massachusetts.

J. R. Finkel and C. D. Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of NAACL HLT*, pages 326–334, Boulder.

Y. Freund and R.E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, volume 904 of *LNCS*, pages 23–37. Springer.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explorations*, 11(1).

A. Hassan and D. Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the ACL*, pages 395–403, Uppsala, Sweden.

H. Kim and J. Kim. 2004. Combining active learning and boosting for naïve bayes text classifiers. In *Advances*

[4] http://phpnuke.org/files/gpl.txt

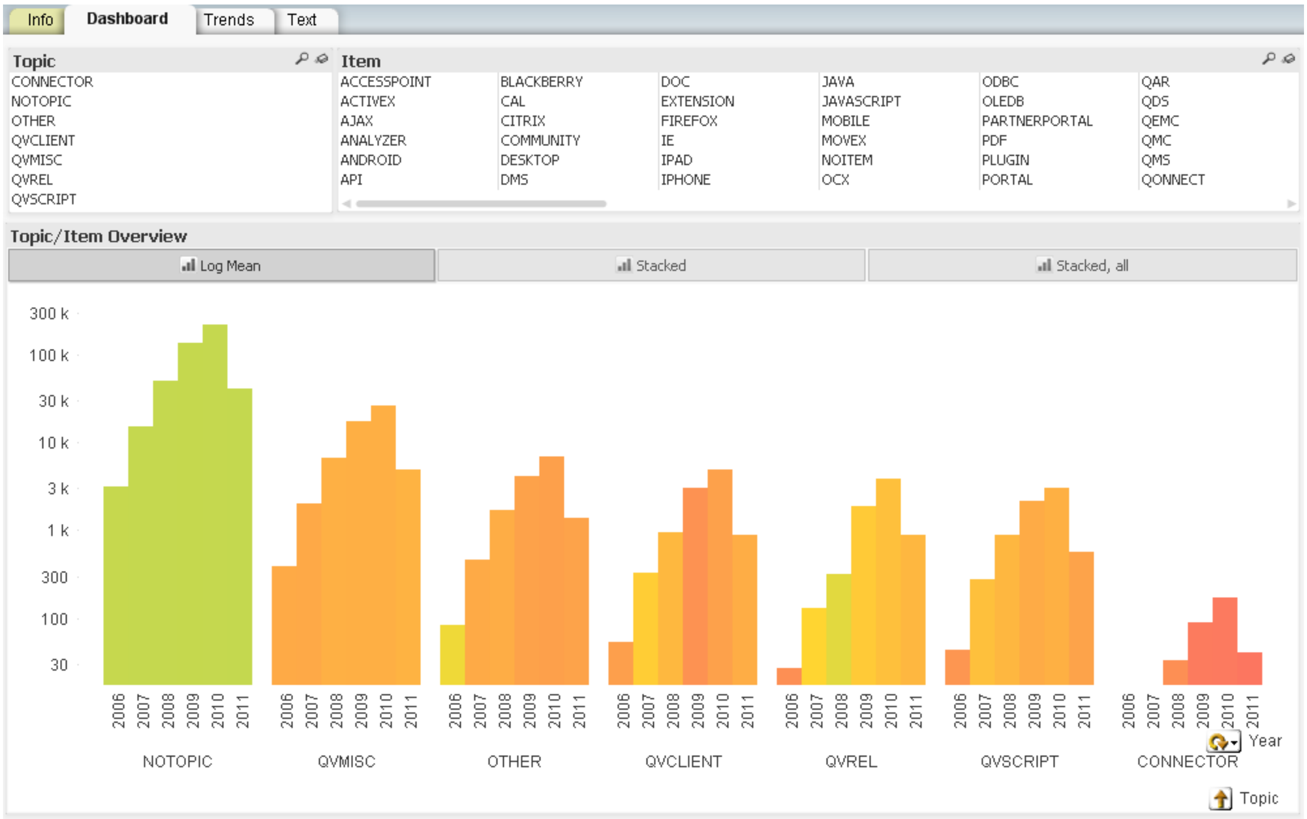
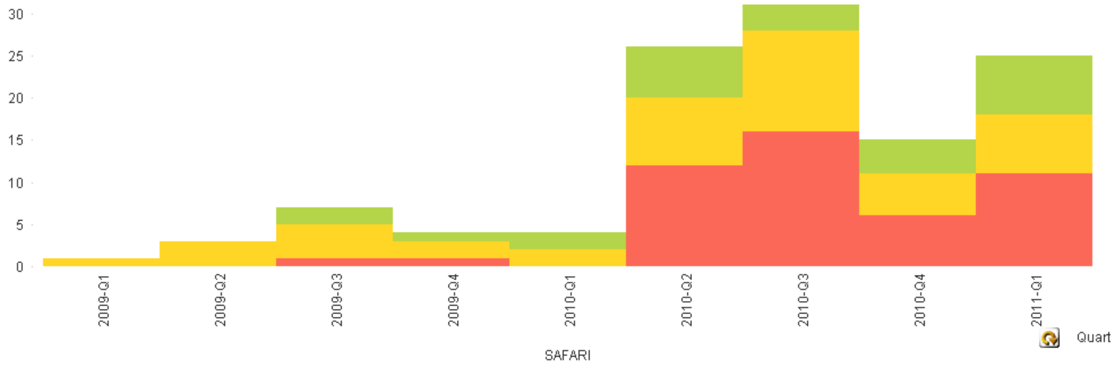
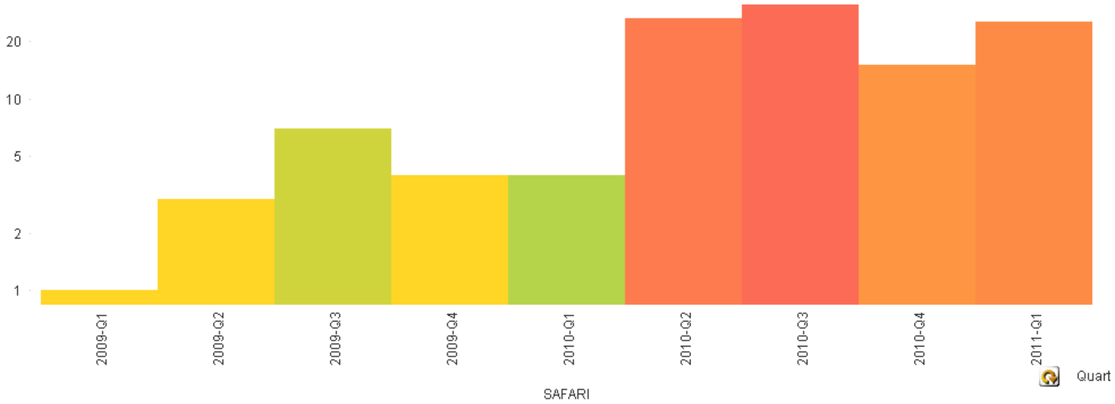


Figure 3: The main dashboard showing the opinion expressed on seven forums.



(a) Visualization of the sentiment expressed in one forum, stacked values: green means positive, red negative, and yellow is neutral



(b) Visualization of the sentiment expressed in one forum, averaged color values.

Figure 4: Visualization of the sentiment expressed in one forum.

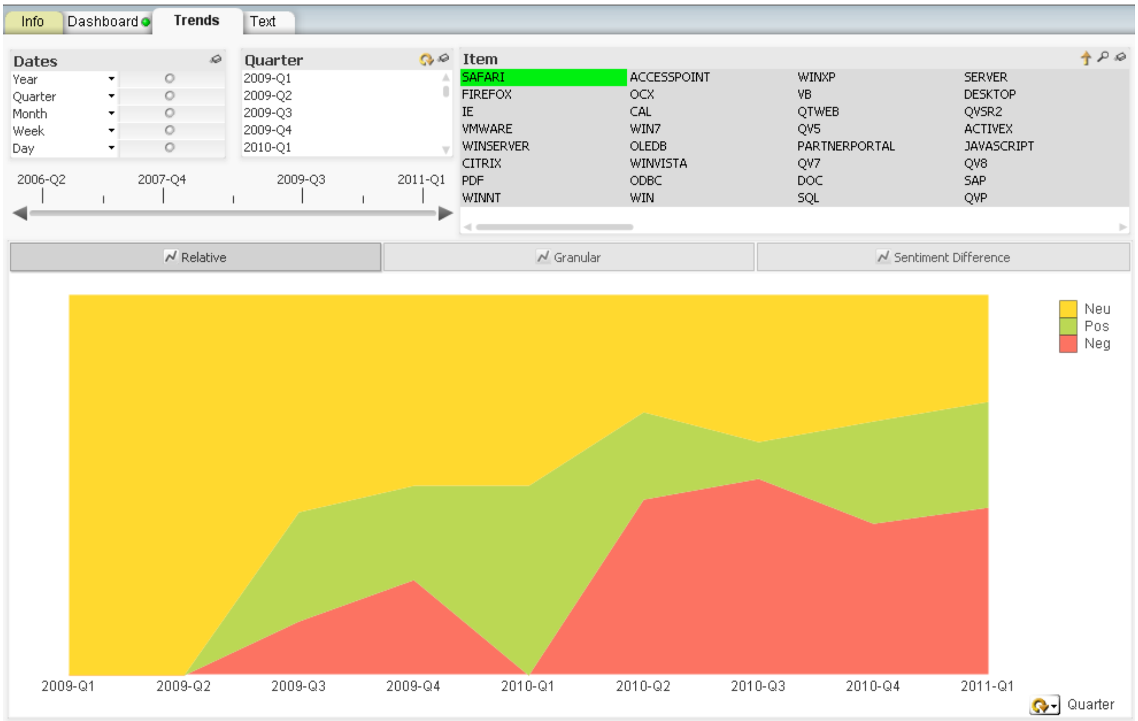


Figure 5: The trend dashboard.

in Web-Age Information Management, volume 3129 of *LNCS*, pages 519–527. Springer.

D. Klein and C.D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the ACL*, pages 423–430, Sapporo.

G.A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

T. Nakagawa, K. Inui, and S. Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of NAACL HLT*, pages 786–794, Los Angeles.

B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, January.

M. Rosell and V. Kann. 2010. Constructing a swedish general purpose polarity lexicon random walks in the people's dictionary of synonyms. In *Proceedings of SLTC 2010*, pages 19–20, Linköping.

O. Täckström and R. McDonald. 2011. Discovering fine-grained sentiment with latent variable structured prediction models. In *Proceedings of ECIR*, Dublin.

S.B. Wang, A. Quattoni, L.P. Morency, D. Demirdjian, and T. Darrell. 2006. Hidden conditional random fields for gesture recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1521–1527, Los Alamitos, CA, USA.

J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(1-2):165–210.

T. Wilson, J. Wiebe, and P. Hoffman. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP*, pages 347–354, Vancouver.

T. Zagibalov, K. Belyatskaya, and J. Carroll. 2010. Comparable English-Russian book review corpora for sentiment analysis. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 67–72, Lisbon.