RENDER
FP7-ICT-2009-5
Contract no.: 257790
www.render-project.eu

RENDER

Deliverable D2.1.1

Prototype of the Opinion Mining Toolkit

Editor:	Delia Rusu, JSI
Author(s):	Delia Rusu, JSI; Blaz Fortuna, JSI; Mitja Trampus, JSI; Andreea Bizau, JSI; Tomaz Hocevar, JSI
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	September 2011
Actual Delivery Date:	September 2011
Suggested Readers:	developers working on WP4 – Diversity Toolkit, developers creating case study prototypes - WP5
Version:	1.0
Keywords:	opinion mining, sentiment analysis, polarity, active learning, bias detection, viewpoints

Full Project Title:	RENDER – Reflecting Knowledge Diversity
Short Project Title:	RENDER
Number and Title of Work package:	WP2 Diversity mining
Document Title:	D 2.1.1 – Prototype of the Opinion Mining Toolkit
Editor (Name, Affiliation)	Delia Rusu, JSI
Work package Leader (Name, affiliation)	Delia Rusu, JSI

Executive Summary

The Prototype of the Opinion Mining Toolkit deliverable is comprised of two main parts: a first part dedicated to sentiment analysis, and a second one to detecting bias by first detecting viewpoints or aspects expressed in a corpus.

The sentiment analysis task is defined as automatically determining the polarity of words – positive, negative or neutral in a given corpus. The task is approached from two different perspectives: firstly, by identifying sentiments with the aid of a domain-driven sentiment vocabulary; the second approach relies on employing active learning techniques for sentiment and topic analysis.

Our algorithm for generating a domain-driven sentiment vocabulary was published as a workshop paper at DiversiWeb 2011, and is annexed to this deliverable.

In the case of bias detection, we perform a first step and propose a novel approach to automatic viewpoint identification using a multilevel k-means clustering algorithm.

The results we obtained so far in all three main research directions presented in the deliverable are motivating. Firstly, in the case of the domain-driven sentiment vocabulary, our method can identify more domain-specific terms compared to the more generic SentiWordNet baseline. However, we still need to perform a qualitative analysis of the words for which the negative and positive polarity was assigned. Secondly, for topic and sentiment identification based on active learning we show evaluation results using the well-known precision and recall evaluation metrics, as well as the learning curves, using a dataset including tweets in both English and Spanish. In the viewpoint detection case, while we see that the method is still lagging compared to the state of the art, possibly due to it making fewer assumptions about the data, it clearly improves over the baseline.

As future work, we plan to combine the active learning approach with the domain-driven sentiment vocabulary work, which should result in a higher performance of the sentiment models. We are also going to further pursue our work on bias detection and cross-lingual opinion analysis. Additionally, we plan to incorporate the functionality from the active learning command-line utility, so that it will be available over a web interface.

Table of Contents

List of Figures

Abbreviations

DiversiWeb	First International Workshop on Knowledge Diversity on the Web (DiversiWeb 2011), co-located with the 20th World Wide Web Conference (WWW 2011). Hyderabad, India
IMDb	Internet Movie Database
SentiWordNet	a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity.
US	uncertainty sampling
BOW	bag-of-words
API	Application Programming Interface
LDA	latent Dirichlet allocation
TAM	Topic-Aspect Model
SVM	Support Vector Machines
TREC	**T**ext **RE**trieval **C**onference

Definitions

Opinion	a quadruple [Topic, Holder, Claim, Sentiment] in which the Holder believes a Claim about the Topic, and can associate a Sentiment to the belief
Opinion mining	identifying opinions using machine learning techniques
Sentiment	expression in text (explicit or implicit) stated by the Holder. The expressions can have a positive, negative, or neutral regard toward the Claim about the Topic.
Polarity	it refers to sentiment polarity, whether it is positive, negative or neutral
Bias detection	given a topic, event, entity, etc, the task of bias detection is to determine the differences between a target object of analysis and another one. This object of analysis can be, for e.g. the author/publisher of a news article.
Support vector machine	it is a set of related supervised learning methods for performing data analysis and pattern recognition, used for classification and regression analysis.
k-means ++	is an algorithm for choosing the initial values (or "seeds") for the *k*-means clustering algorithm. It is an approximation algorithm for the NP-hard *k*-means problem.

1 Introduction

According to the terminology used within the RENDER project, and formally defined in D3.1.1 [29], the opinion is expressed by an opinion holder with respect to a certain topic, and can have an associated sentiment (such as *good* or *bad*).

For defining an opinion, we adopt Kim and Hovy's [24] definition, which states the following: an **opinion** is a quadruple [Topic, Holder, Claim, Sentiment] in which the Holder believes a Claim about the Topic, and can associate a Sentiment to the belief. A **sentiment** is an expression in text (explicit or implicit) stated by the Holder. The expressions can have a positive, negative, or neutral regard toward the Claim about the Topic.

The prototype of the Opinion Mining Toolkit is mainly focused on sentiment analysis, by automatically determining the polarity of words – positive, negative or neutral. A second focus is towards bias detection in a corpus, with the aid of viewpoints.

The deliverable includes the following three main sections: a first section describing an approach to building a domain-driven sentiment vocabulary, a second section on sentiment and topic analysis from an active-learning perspective, and a final section on detecting bias by identifying various viewpoints or aspects expressed in a corpus.

Our algorithm for generating a domain-driven sentiment vocabulary was published as a workshop paper at DiversiWeb 2011, and is annexed to this deliverable (see Annex A.1). We describe the steps towards building the sentiment vocabulary, and refer to preliminary results obtained when using a vocabulary built from movie reviews to analyse tweets related to movies.

For the active learning approach to sentiment and topic analysis we considered a dataset formed of tweets represented using a bag-of-words model. We defined the task of determining the topic and sentiment of a given tweet as a classification problem. We show results for topic and sentiment identification using the well-known precision and recall evaluation metrics.

In the case of bias detection, we perform a prerequisite step and propose a novel approach to automatic viewpoint identification. Based on a finding that clusters are far more likely to correspond to topics rather than viewpoints, we transfer the two-level approach to modelling viewpoints from latent Dirichlet allocation (LDA)-like to k-means clustering. This has two main potential advantages: *speed*, as the complexity of a single iteration of k-means is linear in the number of documents and the number of iterations tends not to be large in practice; the second advantage is *similarity-based modelling* - graphical models view each document as a result of some generative process and ultimately represent it as a collection of draws from a multinomial distribution. This corresponds to the bag-of-* model, typically BOW.

In the following sections of the deliverable we are going to detail each of the aforementioned points. We start with the work on domain-driven sentiment analysis, listing related work, the approach itself and referring to preliminary results. The next section describes the active learning approach to sentiment and topic analysis, by starting with a short introduction to active learning, presenting the experimental setting and discussing results. The final section presents our work on bias detection based on viewpoint identification.

1.1 Motivation

The presented work is focused on the development and evaluation of the opinion mining technology (currently a prototype) used to support case studies within the RENDER project. Both Telefonica and Wikipedia case studies have highly specific datasets, on which the opinion mining technology is planned to be applied. However, developing good sentiment models requires domain dependent features (vocabulary) and training data. To this end, we approached this in two ways. In Section 2 we focus on developing algorithms for extracting a high-quality domain vocabulary, and in Section 3 we focus on developing algorithms for cost-efficient acquisition of training data. Finally, in the last section of this deliverable we introduce a novel method for viewpoint detection which is completely distance-based. This is relevant for the RENDER project for two main reasons: firstly, one key focus of the project is also predicate-based data representation and analysis; secondly, we consider viewpoint detection as a first step towards bias detection.

2 Domain-driven Sentiment Analysis

For the sentiment analysis task, we considered two different approaches – one based on a domain-driven sentiment vocabulary that we constructed, and another one based on active learning. In this section of the deliverable we are going to further describe the first approach, i.e. domain-driven sentiment vocabulary construction.

2.1 Domain-driven Opinion Vocabulary

A broad overview of the research areas of opinion mining and sentiment analysis is given by Pang and Lee's 2008 survey [21], as well as Liu's book chapter on sentiment analysis and subjectivity [20]. Our workshop paper in Annex A.1 briefly describes related work from the area of sentiment vocabulary construction (see Section 2 of the paper).

We propose a method to construct a sentiment vocabulary by expanding a small set of initial (seed) words with the aid of connectives. The method consists of four steps:

1. Given a positive word seed list and a negative word seed list and making use of WordNet's synsets, we expand the initial seed lists based on the synonym / antonym relations.
2. From a corpus of documents, we parse and extract all adjectives and conjunctions, constructing a set of relationships between the determined words.
3. The third step implies cleaning the resulting set of words and relationship graph by removing stop words and self-reference relations.
4. In the fourth step, we determine the polarity of the words extracted from the corpus by applying an algorithm on the relationship graph obtained in the previous steps.

Expanding the Initial Seed List

The initial words will be assigned a score of 1 for positive words and -1 for negative words, respectively. We compute the polarity score for each newly found word by recursively processing the synsets for each seed word. A word can be found in synsets corresponding to different seed words, either in synonym or antonym relations. Another factor we take into account is the *distance* between the seed word and the currently processed word, as provided by the WordNet hierarchy. From these two considerations, a more formal way to compute the score of a word (s_w) to be added to the seed list is:

$$s_w = \max\left(abs(s_{w,o}) \cdot sign(\max(s_{w,o})\right)$$

where

$$s_{w,o} = \begin{cases} f \cdot s_o, when\ w\ and\ o\ are\ synonyms \\ -f \cdot s_o, when\ w\ and\ o\ are\ antonyms \end{cases}$$

and o is a seed word, while f is a parameter for which we empirically assigned values between 0 and 1; further experiments need to be carried out to motivate our empirical findings. The result of this step is an expanded seed word list together with their polarity score.

Extracting Adjectives and Conjunctions

There can be two types of relationships, indicating if two or more words have the same context polarity (words connected by *and*, *or*, *nor*) or opposite polarity (words connected by *but*, *yet*). We will refer to them in the following algorithms as *ContextSame* and *ContextOpposite* relations, respectively.
Based on the determined relations, we can then construct a relationship graph *G(W, E)*, where

- *W={set of determined adjectives}* and
- *E={*w_iw_j, where w_i, w_j from *W* if there is a determined relationship between w_i and w_j, each edge having a positive weight for the *ContextSame* relationship and a negative weight for the *ContextOpposite* relationship}.

In what follows, we describe the algorithm for building the relationship graph G (see Figure 1).

```
G = ({}, {})
foreach document d in corpus
    foreach sentence s in d
        parseTree = GetParseTree(s)
        {w,c} = RetrieveWordsAndConjunctions(parseTree)
        ConstructRelationGraph(G, {w, c})
        HandleNegation(G, s)
```

Figure 1. The algorithm for constructing the relationship graph G.

We used a maximum entropy parser[1] to retrieve a sentence's parse tree that we then analyse in the *RetrieveWordsAndConjunctions* procedure. We construct an adjective stack *w* and a conjunction stack *c* by extracting the relevant nodes according to their part-of-speech tags and group them together based on the common parent node between the adjective nodes and the conjunctions nodes. In the *ConstructRelationGraph*, we will add the nodes for each newly found adjective and add new edges to the relationship graph *G* according to each conjunction's behaviour. Each edge has an associated weight with values between 0 and 1, determined by optimization. We handle the presence of negation in the sentence by reversing the type of the relation, if a negation is detected. For example, considering the sentence "*Some of the characters are fictitious, but not grotesque*", the initial relation between *fictitious* and *grotesque* would be a *ContextOpposite* relationship, but the presence of the negation is converting it to a *ContextSame* relationship.

Determining the Polarity of Words

In the fourth step, we determine the polarity of the words extracted from the corpus by applying an algorithm on the relationship graph obtained in the previous steps, which was inspired by the well-known PageRank algorithm [1]. For this, we define two score vectors, a positivity score *sPos* and a negativity score *sNeg*, respectively. We choose the final score to be the sum of the positivity and negativity score. The sign of the score represents the word's polarity, that is, a positive score characterizes a positive sentiment polarity, while a negative score characterizes a negative polarity. The algorithm is presented in Figure 2, and described in what follows.

```
InitializeScoreVectors(sPos(W), sNeg(W))
do {
    foreach word w_i in W
        foreach relation rel_ij in relationship graph G that contains w_i
            if rel_ij is a ContextSame relation
                sPos(w_i) += weigth(rel_ij) * prevSPos(w_j)
                sNeg (w_i) += weigth(rel_ij) * prevSNeg(w_j)
                else if rel_ij is a ContextOpposite relation
                sPos(w_i) += weigth(rel_ij) * prevSNeg(w_j)
                sNeg (w_i) += weigth(rel_ij) * prevSPos(w_j)
        NormalizeScores(sPos(w_i), sNeg(w_i))
} while more than 1% of the words w_i in W change orientation
```

Figure 2. The algorithm for determining the polarity of words extracted from a corpus.

[1] http://sharpnlp.codeplex.com/

We initialize the score vectors based on the polarity scores of the expanded seed word list (see step 1). We will assign the corresponding positivity or negativity score sw_j for each adjective w found in the seed list. For the opposite score we assign a very small value (ε), in order to allow for meaningful values when computing the score for *ContextOpposite* relations.
A *ContextSame* relation enforces the existing positive and negative scoring of w_i proportionally with the scoring of w_j. A *ContextOpposite* enforces the negativity score of w_j with respect to the positivity of w_i, and the positivity score of w_j with respect to the negativity score of w_i.

2.2 Results

As this work was carried out early during the project (first 6 months), we conducted our experiments on a publicly available movie review corpus. Our aim was to see how well a domain specific vocabulary constructed from movie reviews performs when applied to analysing tweets. We used a document corpus of 27,886 IMDb (Internet Movie Database) movie reviews[2] and constructed a movie domain specific vocabulary according to the aforementioned approach. We retrieved 9,318 words, from which 4,925 have a negative polarity and 4393 have a positive polarity, starting from a small seed list of 10 positive and 10 negative adjectives. We refer the reader to Annex A.1 for more examples of words from the domain-specific vocabulary.

For our tests, we crawled 220,387 tweets, using the Twitter Search API[3], over a two month interval, keyed on 84 movies, spanning different genres and release dates. Without actually classifying each tweet, we counted the frequency of positive and negative sentiment words that we identified in the collection of tweets. To resume our findings, which are presented in more detail in **Fehler! Verweisquelle konnte nicht gefunden werden.**Annex A.1, we observed a relationship between our score obtained by counting the positive sentiment words and the IMDb score. In our future work we plan to conduct a series of experiments in order to determine if there exists a correlation between the two numbers: the IMDb rating and the number of positive sentiment words.

2.3 Prototype

The domain-driven sentiment vocabulary prototype was implemented as a desktop application[4] and requires Microsoft Windows to run. The application has two main parts:

1. A domain-driven sentiment vocabulary generation part
2. A Twitter analysis part

In the first part, the sentiment vocabulary is created. As input, the application requires a list of seed words as well as a document collection. The document collection is required to build the sentiment vocabulary, starting from the seed words. The content of the vocabulary can be visualized as a list or as a graph, and compared with SentiWordNet.

In the second part, a sentiment vocabulary is used to analyse tweets belonging to the same domain as the one used to build the vocabulary from. As input, a collection of tweets belonging to a specific domain, and preferably from a certain topic is required. The application identifies vocabulary words in the tweet collection, assigns sentiment polarity scores based on the vocabulary scores and offers several visualizations for analysing tweets.

The default dataset for the application belongs to the movie domain.

2.4 Conclusions

We have compared our results to SentiWordNet, a state-of-the-art sentiment vocabulary, which is not domain specific. Quantitatively, our method retrieves, for the aforementioned dataset, more sentiment-

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[3] http://search.twitter.com/api/
[4] http://km.aifb.kit.edu/projects/render/index.php/OpinionMiningPrototype

bearing adjectives than provided by SentiWordNet. SentiWordNet contained only 5,056 of the 9,318 retrieved words. By inspecting the results, we determined that this is due mainly to the domain-specific words. We still need to perform a qualitative analysis of the words for which the negative and positive polarity was assigned.

Concerning our task at hand, that is sentiment analysis on tweets related to movies, we cannot directly compare our results to other approaches, as there is (to the best of our knowledge) no standardized dataset. We did compare our positive movie score to the IMDb movie score for around 10 movies in our dataset, with promising results. However, for statistical significant results we have to collect data for more movies.

3 Active Learning for Sentiment and Topic Analysis

3.1 Active Learning

Active learning [18] is a generic term describing an interactive learning process. In the usual supervised learning the learning method is presented with a static training set that is used to construct a model. The active learning paradigm assumes the availability of an unlabelled set, from which the learning method ('student') can select few examples and 'asks' the 'oracle' (e.g. a domain expert, the user) to label them [18]. The new labelled examples are used to update the model, closing the active learning loop (Figure 3).

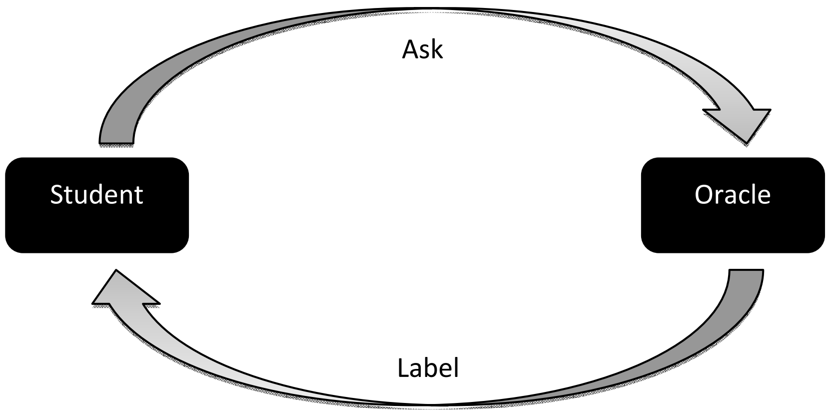

Figure 3. The active learning loop: the student selects examples and asks the oracle to label them. The oracle provides labels back to the student, who uses them to update the model and select the next round of examples to label.

The questions (examples to be labelled) are selected based on their potential to improve the existing model if their labels are known. The goal of the process is to approach the accuracy of a model, trained on completely labelled dataset, but with as few labelled examples as possible. There are different methodologies for selecting examples, uncertainty sampling being one of the most popular.

The visualization in Figure 4 shows three clusters: *finance*, *car manufacturing*, and *information technology*. The visualization shows how the financial companies are isolated through sampling of the space and how technology companies, not in the initial training set, are then being selected as most informative for determining the space. Instances from the training set are marked as bold and the instance in question is marked as italic. The classification model is depicted as a separating line, with the instance closest to the line being selected as the question.

3.2 Application to Sentiment and Topic Analysis

Sentiment analysis techniques require domain dependent training data. For example, sentiment models developed on movie reviews are not directly applicable to other domains, due to domain dependent vocabularies typically used to express sentiments. Also, a given document, knowledge of its domain is required before an appropriate sentiment model can be applied.

In this section we will focus on extraction of sentiments from tweets. Given a set of twitter messages (tweets), the task is to extract tweets expressing positive and negative sentiments on a given topic. This is done in two stages. First, we extract tweets which are relevant to our topic and then we partition the relevant ones based on their sentiment.

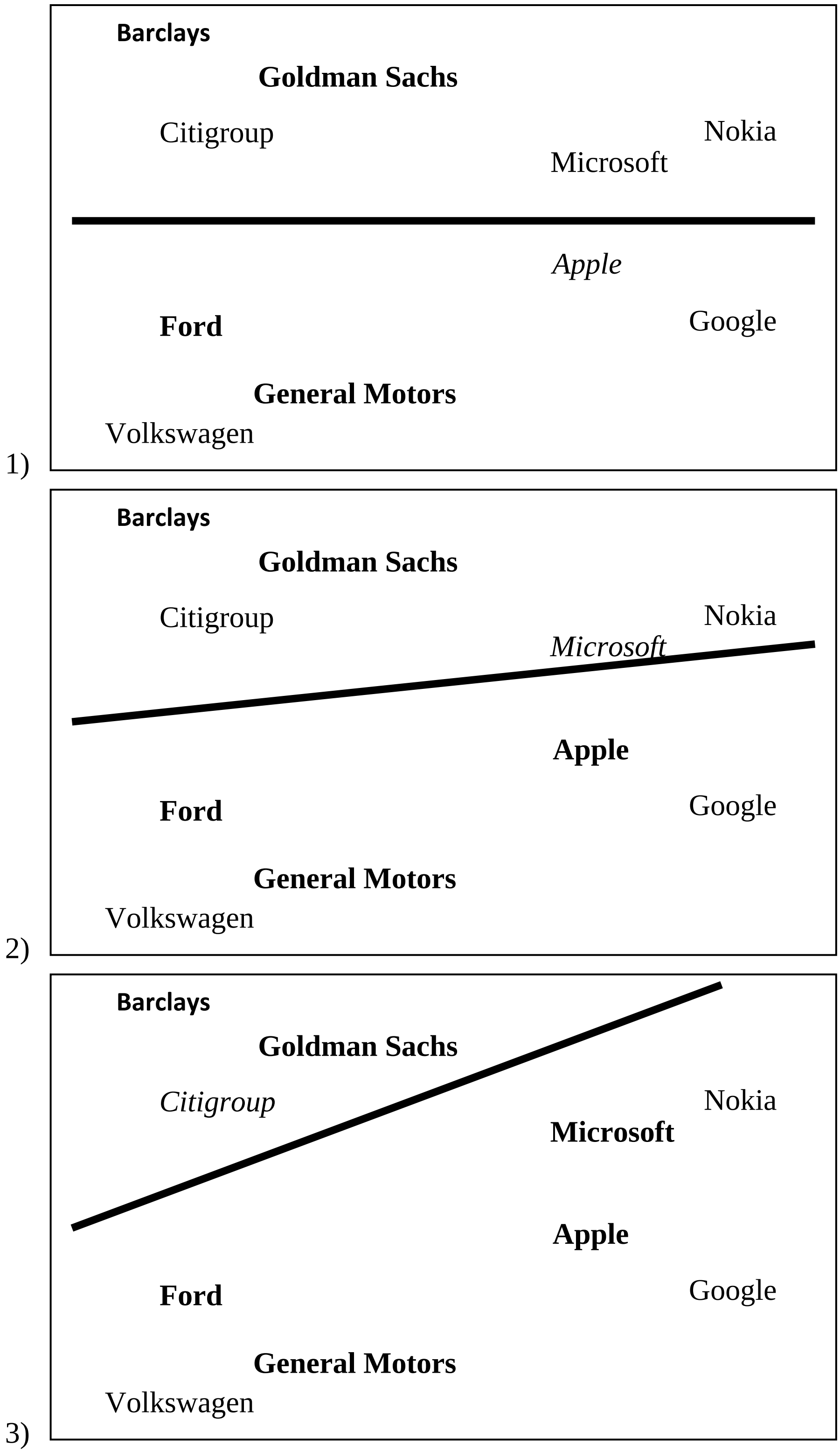

Figure 4: Visualization of the active learning loop.

To reduce the cost of training data acquisition we investigated the use of active learning in the above task. To train a classifier we usually need a training set which consists of some input instances and their correct classifications. The size of the training set depends on the difficulty of the underlying model which we are trying to learn and on the required accuracy of the classifier. Active learning aims to reduce the size of this training set. There is almost unlimited amount of tweets but the task of classifying them to create a training set is very time consuming. Active learning incorporates the selection of training instances into the learning process.

On each step of the learning process, the learner selects one or more instances such that their correct classification would provide the most insight to the learner. There are many strategies how to choose these instances. We used uncertainty sampling strategy (which is one of the most popular strategies), which selects instances for which the currently built classifier is the most uncertain how to classify them. In case of binary classification task this method selects instances with predicted probabilities closest to 0.5. The method was selected since it was proven to provide good results and its potential to scale to large datasets [18].

We use bag-of-words (BOW) [19] representation for tweets and perform classification with Naïve Bayes [19] classifier in both topic and sentiment classification stages. Naïve Bayes assumes that the presence of some feature of a class is independent of any other features. In combination with BOW representation this means that each word contributes towards or against the class independently of other words in a tweet. This can be amended by using a bag of n-grams instead of individual words but it requires a larger training set.

3.3 Results

We performed experiments on a set of almost 4 million tweets which were obtained in shorter intervals over 3 days. The chosen topic for classification was Tour de France which was in progress at the time. There are many words which are specific for this topic (e.g. names of teams and cyclists) as well as some confusing ones. Words such as tour and stage could be indicative of our topic or they could be related to some rock band. Sparseness of positive examples presents another challenge. To maintain a good balance of positive and negative examples in the training set, we biased the uncertainty sampling towards the positive class. Instances selected for classification were those with predicted probabilities around 0.6.

We began the experiment with a seed of 2 positive and 2 negative examples. After classifying 50 instances which were queried by active learner, we evaluated the accuracy of such a classifier. All unclassified tweets were divided into 10 buckets based on their predicted class probability (Figure 5). The same was done after 100, 200 and 400 classified learning instances.

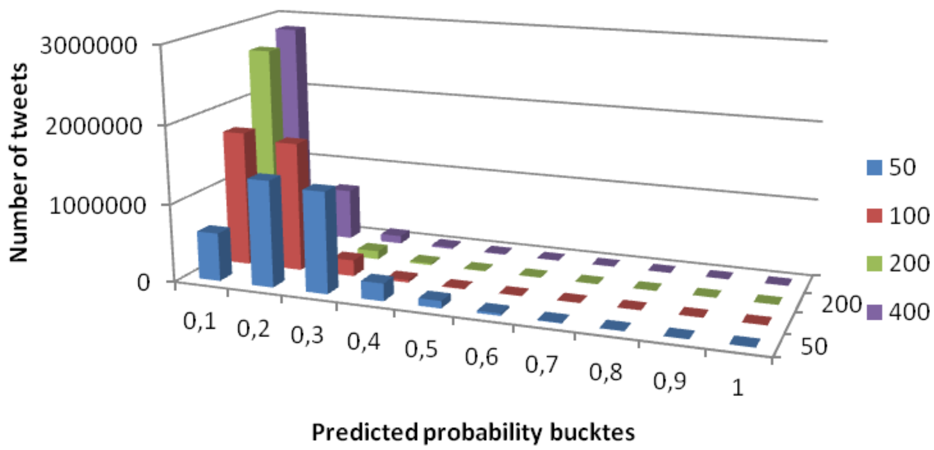

Figure 5. Distribution of tweets by predicted probabilities.

A sample of 100 tweets was randomly chosen from each bucket. We employed crowdsourcing to classify the resulting 4000 tweets (1000 for each of the 4 active learning steps at 50, 100, 200 and 400 tweets). Each tweet was classified twice and any disagreements were checked again by hand. This gave us a rough estimate of precision and recall in each bucket. Figure 6 shows F-score at different probability bounds between negative and positive examples of the topic. The results after 100 and 200 steps of active learning are clearly better than after 50 but the decline at 400 steps was unexpected. The recall increased by 10%

but precision dropped by 20%. This was most likely due to many new words which were introduced in the last 200 queries of active learning.

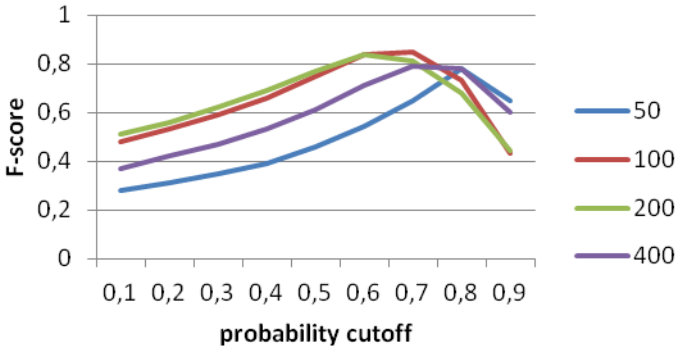

Figure 6. F-score for different predicted probability thresholds

The F score [25] (also referred to as F-measure) is a measure that trades off precision versus recall. It is computed as a weighted harmonic mean of precision and recall:

$$F = \frac{(\beta^2+1)PR}{\beta^2 P+R}, \beta^2 = \frac{1-\alpha}{\alpha}, \alpha \in [0,1], \beta^2 \in [0,\infty] \quad (1)$$

The balanced F measure equally weights precision and recall, i.e. $\alpha = \frac{1}{2}\ or\ \beta = 1$

$$F_{\beta=1} = \frac{2PR}{P+R} \quad (2)$$

The F-score in equation (2) is the one used in this deliverable.

The sentiment analysis task was considered as classification into three categories: neutral, positive and negative. For this purpose we used three binary Naïve Bayes classifiers. They were trained in the same way as for the topic classification task, i.e. through active learning with uncertainty sampling. We classify an instance into the category which has the highest predicted probability according to the classifier for that category.

The experiment was performed on a set of 120 positive, 180 negative and 200 neutral tweets. Non-neutral tweets belonged to two different topics and were in Spanish and English. Neutral tweets were all in English. After each step of active learning we measured the classification accuracy and compared the uncertainty sampling strategy to random sampling. The entire experiment was repeated 100 times. In this setting the uncertainty sampling performed slightly better than baseline random sampling as displayed in Figure 7.

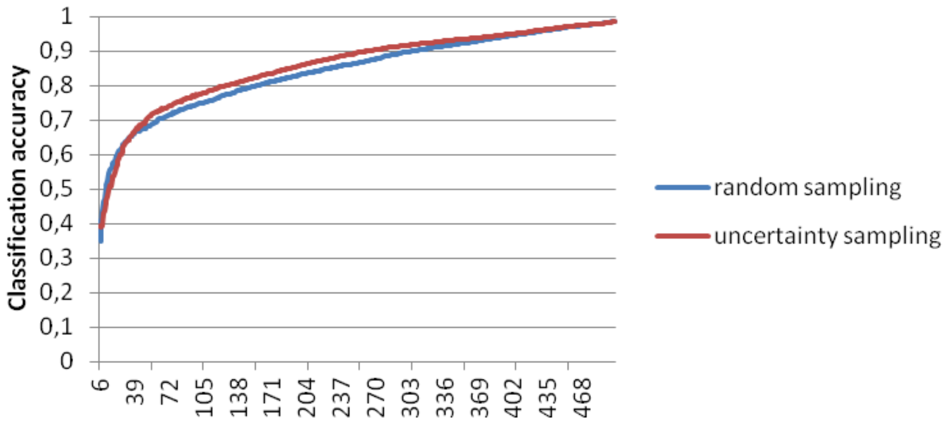

Figure 7. Learning curve

3.4 Prototype

The tested active learning approach was implemented and is available in a form of a command-line utility [5]. The utility can be used to either filter the provided set of tweets by topic or categorize them based on their sentiment. The process in both cases is kicked-started by providing a set of seed tweets. The seed tweets are used to train an initial Naïve Bayes model, which is than further developed through the active learning loop.

The active learning utility can be used to filter tweets by topic as follows. First, a file with a complete set of tweets, a file with positive (topic-related), and a file with negative (not related to topic) tweets are provided as input. The utility then guides the user through the active learning loop, asking for manually classifying a number of selected tweets. Figure 8 provides an example of a short active learning trail. The result is a file with all the positively classified tweets stored in the ***topic.txt*** file.

The utility can be used in a similar way to classify a given set of tweets for sentiment. However, three sets of seed examples are required: of positive sentiment, of negative sentiment, and neutral tweets. The seed examples are preferably selected from the same topic, as the set of tweets to be classified.

The utility is available from the internal Wiki of the RENDER project. The utility is implemented on top of the TextGarden[6] library, and requires Microsoft Windows to run.

3.5 Conclusions

In this section we tackle the problem of learning sentiment and topic models for classifying tweets. We introduce the active learning approach, which can be used for more efficiently preparing the training data, which is required to train sentiment and topic models. A two stage approach is chosen, where:

1. the target topic is extracted from a large collection, and
2. the sentiment models are trained specifically for the target domain.

The approach was tested on a popular topic (Tour de France), chosen due to the familiarity of the general public with it. This enabled us to evaluate the developed algorithm using a crowdsourcing service. Finally, sentiment models were tested on a manually assembled test set, provided by Telefonica. The goal was to test the capability of the models to capture features required for sentiment classification.

[5] http://km.aifb.kit.edu/projects/render/index.php/OpinionMiningPrototype
[6] http://ailab.ijs.si/text-garden-tools/


```
>nbc.exe -in:topic_in.txt -pos:topic_pos.txt -neg:topic_neg.txt

Naive Bayes - Active Learning

===========================================
Sentiment-Classifiaction (-senti:)=No
Input-File (-in:)=topic_in.txt
Positive-Seed-File (-pos:)=topic_pos.txt
Negative-Seed-File (-neg:)=topic_neg.txt
Neutral-Seed-File (-obj:)=
Load-State-File (-state:)=
Load-Only-Bow-From-State-File (-bow:)=
Uncertainty-Sampling (-uncert:)=0.5
=============================

Topic classification

WHAT A STUNNING DAY FOR SOME PROPER ROAD RACING ROUND THE SOUTHERN 100 COURSE ON THE IOM! LET'S HOPE WE HAVE SOME MORE CRACKING RACES! #IOM

[p]ositive/[n]egative/[b]reak: n

THEEEEEN I COULDA SWORN I WAS IN MARIO KART WHEN I WAS ON THE 110NORTH

[p]ositive/[n]egative/[b]reak: n

YESTERDAY MY GRANDMOTHER TURNED 80...IT REALLY PUT WORLD HISTORY INTO PERSPECTIVE FOR ME.

[p]ositive/[n]egative/[b]reak: n

VOECKLER IN THE LEAD: WHERE HAVE WE SEEN THIS BEFORE?: TOUR DE FRANCE MOUNTAIN MEN GET CHANCE TO SHINE - YAHOO├óΓé¼┬ª (CONT) HTTP://DECK.LY/~LMAR3

[p]ositive/[n]egative/[b]reak: p

@ALANPETERSON YOU TOURED WITH FLEETWOOD MAC?  THAT'S A CRAZY RIDE

[p]ositive/[n]egative/[b]reak: n

OKAY, I NEED TO GET READY TO GO TO THE TOUR DE FRANCE NOW.

[p]ositive/[n]egative/[b]reak: p

CANT BELIEVE I AM UP AT 4 AM TO WATCH THE TOUR. BUT IT IS WHAT BIKE GEEKS DO.

[p]ositive/[n]egative/[b]reak: p

YOU'RE MY HERO @DANIELTOSH, "YOU'RE SLOWER THAN A HERD OF TURTLES STAMPEDING THROUGH PEANUT BUTTER." #MARRYME #HILARIOUS #BESTQUOTEEVER

[p]ositive/[n]egative/[b]reak: b

Classifications saved to topic.txt, nottopic.txt
```

Figure 8. Example of using active learning utility

4 Bias Detection

We also performed some early experiments on the topic of bias detection. In this deliverable we started to analyse bias by first identifying viewpoints (also called aspects) expressed in a corpus. A viewpoint can be represented by a list of representative documents (=clustering), sentences (= viewpoint-aware summarization) or keywords.

In some corpora, e.g. product reviews, this corresponds quite well to the better-researched task of sentiment analysis. On others, e.g. political debates, there is no clear "positive" and "negative". Despite that and assuming there are only two major opposing viewpoints present in the corpus, our task is seemingly still very similar to that of sentiment detection: instead of a "positive" and "negative" view on a product, we have e.g. a "Palestinian" and "Israeli" view on a political situation. The key difference is, however, that in the sentiment case, the two viewpoints are known in advance. This makes it possible to associate scores with every word based on how indicative it is of each class. The scores can be computed in advance (see SentiWordNet) or induced later on in a domain-specific way, but even then typically with a seed list of domain-independent words like "good", "excellent", "terrible" and "bad". Even if such a list is not employed, the algorithms can relatively safely assume that opposing views will be, to a large degree, expressed with opposing adjectives. Those can be identified either by appearing in specific syntactic constructs (e.g. "but" sentences, "fast but expensive", as somewhat famously proposed by McKeown [2]) or by using a background knowledge source (e.g. thesauri).

In contrast, more complex opinions are less likely to be expressed simply with adjectives; finding opposing pairs of statements it therefore harder (and sometimes objectively impossible).This intuition is supported by Lin and Hauptmann [9]. Working on two different datasets , they find that the Kullback-Leibler (KL) divergence between bag-of-words (BOW) distributions induced from document collection pairs is an order of magnitude smaller between aspects than between different topics.

4.1 Related work

Related to our work is the TREC 2002-2004 Novelty Track [15]. The task there is first to identify sentences relevant to a query, which is essentially a passage retrieval task, and then to identify those relevant sentences that contain *novel* information, i.e. information that has not appeared previously in the topic's set of documents. Although the setting is different from ours, it is equally important to identify similar and differing facts. The participants worked at the sentence level and in the majority of cases used the BOW representation and vector space models, usually with the cosine distance. Two teams used WordNet expansion of terms, but its impact on results was not significant. Parsing of sentences was not used.

In a similar vein, the textual entailment task is also related to our goal, as are question answering and other fields. We only give pointers to a survey article [16] here.

Complementary to our goal of extracting opposing views, there are also attempts at identifying subjectivity of opinions (which we do not presently focus on) rather than their content. Pang [12] has shown that such information can be mildly useful in cases like the document sentiment classification scenario.

Directly pertaining our task, the possibility of using structured semantic information extraction for the purpose of sentiment extraction was suggested relatively early by Lini [11], though it does not seem to have been followed up by an implementation and/or experiments. Similarly, a position paper by Cardie [17] proposes the problem of multi-perspective question answering, focusing on how to comprehensively answer questions where the answer is not factual but rather dependent on personal opinion. Fortuna et al [5] describe viewpoints with keywords learned from the separating plane of a support vector machine (SVM) trained to distinguish between the viewpoints' BOWs; this assumes consistency in topicality across documents. Hardisty et al. [7] make a similar assumption and use a variant of Naïve Bayes for (supervised) viewpoint classification based on BOW and frequent n-grams. Greene et al. [6] perform viewpoint classification with features based on presence of hand-defined patterns in the dependency parse trees of sentences. Most importantly, Paul et al [13] have recently proposed, TAM (Topic-Aspect Model), a graphical model which models words as being drawn from a distribution belonging to a topic (like LDA), an aspect, or a combination of a topic and an aspect (introduced in [14]). They successfully use it both in the supervised and unsupervised setting to achieve state of the art viewpoint clustering accuracy. They further improve this performance by performing dependency parsing on documents and then representing each document as a bag of (object, verb) and (verb, subject) dependencies rather than a bag of words.

4.2 Multilevel k-means Clustering

We propose a novel approach to automatic viewpoint identification. Based on the aforementioned finding that clusters are far more likely to correspond to topics rather than viewpoints and encouraged by Paul's [13] success, we transfer the two-level approach to modelling viewpoints from LDA-like to k-means clustering. This has two main potential advantages:

• Speed. The complexity of a single iteration of k-means is linear in the number of documents and the number of iterations tends not to be large in practice.

• Similarity-based modelling. Graphical models view each document as a result of some generative process and ultimately represent it as a collection of draws from a multinomial distribution. This corresponds to the bag-of-* model, typically BOW.

K-means clustering, on the other hand, deals only with distances/similarities between pairs of points and imposes no constraints on their representation. This is in line with our work on fact extraction and integration of background knowledge in the mining process: a bag-of-facts representation of a document is far too sparse and also does not enable us to exploit the background knowledge by indicating relatedness between certain pairs of facts. A distance-based approach, on the other hand, is much more natural: given two semantically encoded facts and a background ontology, we can easily assign some heuristic "distance" to the pair.

Our approach works as follows. First, we perform k-means on the initial document set. The number of clusters k is set experimentally, depending on the corpus. These clusters are assumed to correspond to topics. On each of the k resulting clusters, we perform a second level of clustering, this time into two clusters. The two clusters are assumed to represent the two viewpoints on the topic.

Since we are interested in viewpoints rather than topics, we now discard the topic clusters and merge the second-level clusters. This is, however, not trivial. Since we do not know to which of the viewpoints each second-level corresponds, there are 2k-1 ways to merge them across topic clusters, where k is the number of topics. Since k tends to be small, we simply evaluate all 2k-1 possible merging options and select the one where clustering quality (see below) of the resulting two clusters is highest.

Since k-means is rather sensitive to the choice of initial centroids, we restart each clustering process 10 times and return the "best" clustering produced. Clustering quality is estimated by cohesion – the sum of distances between all pairs of points in a cluster, divided by the size of that cluster and summed over all clusters. We use the k-means++ [23] procedure to initialize the centroids.

4.2.1 Results

We performed experiments on the bitterlemons.org dataset first presented by Lin [9]. The dataset is a collection of approximately 600 articles from the bitterlemons.org website. The website is dedicated to the Gaza conflict and presents each week two articles written by Palestinian and two articles written by Israeli authors. The nationality of the author is known and is assumed in our experiments to be interchangeable with his or her viewpoint. This is the standard way to interpret this recent and relatively popular dataset ([9], [10], [8], [13]).

We cluster all the documents in the corpus and measure the clustering accuracy – the classification accuracy we would obtain if we treated cluster labels as classification predictions (assuming the correct alignment of cluster labels and class labels, which is not problematic for only two clusters).

As this is work in progress, we have so far only measured the efficiency of our approach on the simple tf-idf weighted BOW model, not utilizing the additional freedom offered by the distance based approach. We run two models: first, as a baseline, we perform simple k-means clustering into two classes. Second, we run the multilevel k-means as described above, with k=6. To account for randomness introduced by random initial centroid placement, we run each model 300 times and report on the mean accuracy:

- baseline (k-means with k=2): 57.9
- multilevel k-means: 62.4
- TAM with BOW (Paul 2010): 68.2

While we see that the method is still lagging compared to the state of the art, possibly due to it making fewer assumptions about the data, it clearly improves over the baseline. The improvement is statistically significant with 99.9% confidence. We plan to further research and attempt to improve this approach in the future.

4.2.2 Conclusions

We introduced a novel method for viewpoint detection which is completely distance-based. This makes it particularly appropriate for application to data represented with semantic facts which normally have to suffer some simplification of data if we wish to use them with distributional or vector-space models. At the same time, predicate-based data representation that plays an important role in RENDER. The logical next step is therefore to find a good distance measure in the space of semantic assertions and apply it to the method described in this deliverable.

5 Conclusions and Future Work

This deliverable was mainly focused on sentiment analysis, by automatically determining the polarity of words – positive, negative or neutral. A second focus was towards bias detection in a corpus, with the aid of viewpoints. In the first section of the deliverable we described an approach to building a domain-driven sentiment vocabulary. A second section was dedicated to topic and sentiment analysis from an active-learning perspective, and a final section on detecting bias by identifying various viewpoints or aspects expressed in a corpus.

Our algorithm for generating a domain-driven sentiment vocabulary was published as a workshop paper at DiversiWeb 2011, and is annexed to this deliverable (see Annex A.1).

The results we obtained so far in all three main research directions presented in the deliverable are motivating. As far as the domain-driven sentiment vocabulary is concerned, we still need to perform a qualitative analysis of the words for which the negative and positive polarity was assigned. Related to the active learning prototype for topic and sentiment detection, we opted for an approach which can accommodate the cross-lingual requirements of the use cases (mainly Telefonica). However, we have to conduct more experiments and further develop this prototype from the cross-lingual point of view. We show evaluation results for topic and sentiment identification using the well-known precision and recall evaluation metrics, as well as the learning curves. In the viewpoint detection case, while we see that the method is still lagging compared to the state of the art, possibly due to it making fewer assumptions about the data, it clearly improves over the baseline.

As future work to be included in the final deliverable on opinion mining, we are going to extend our work on opinion mining beyond sentiment analysis based on polarity identification, as well as continue our work on bias detection. We plan to combine the active learning approach with the feature extraction work presented in Section 2, which should result in a higher performance of the sentiment models. Additionally, we plan to incorporate the functionality from the active learning command-line utility, so that it will be available over a web interface.

In this deliverable we showed preliminary work on a sample Twitter dataset manually assembled by Telefonica. In the next deliverable on opinion mining we are going to extend our bias detection work and include as test cases datasets provided by Wikimedia.

References

[1] Brin, S. and Page, M. Anatomy of a large-scale hypertextual Web search engine. *In Proceedings of the 7th Conference on World Wide Web (WWW)* (1998).

[2] Hatzivassiloglou, V. and McKeown, K. Predicting the semantic orientation of adjectives. *In Proceedings of the 35th Annual Meeting of the ACL* (1997).

[3] Pang, B. and Lee, L. Thumbs up? Sentiment Classification using Machine Learning Techniques. *In Proceedings of EMNLP* (2002).

[4] Turney, P. D. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *In Proceedings of the 40th Annual Meeting on ACL* (2002).

[5] Fortuna, B., Galleguillos, C., Cristianini, N. Detection of Bias in Media Outlets with Statistical Learning Methods. *Text mining: classification, clustering, and applications 10, 27* (2009)

[6] Greene, S., Resnik, P. More than Words: Syntactic Packaging and Implicit Sentiment. *Computational Linguistics pp. 503-511* (2009)

[7] Hardisty, E.A., Boyd-Graber, J., Resnik, P. Modeling perspective using adaptor grammars. *In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 284–292. Association for Computational Linguistics.* (2010)

[8] Klebanov, B.B., Beigman, E., Diermeier, D. Vocabulary choice as an indicator of perspective. *In: Proceedings of the ACL 2010 Conference Short Papers, pp. 253–257. Association for Computational Linguistics* (2010)

[9] Lin, W.H., Hauptmann, A. Are these documents written from different perspectives?: a test of different perspectives based on statistical distribution divergence. *In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 1057–1064. Association for Computational Linguistics* (2006)

[10] Lin, W.H., Wilson, T., Wiebe, J., Hauptmann, A. Which side are you on?: identifying perspectives at the document and sentence levels. *In Proceedings of the Tenth Conference on Computational Natural Language Learning, pp. 109–116. Association for Computational Linguistics* (2006)

[11] Lini, D., Mazzini, G. Opinion classification through information extraction. *Intl. Conf. on Data Mining Methods and Databases* (2002)

[12] Pang, B., Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *Proceedings of the 42nd ACL* (2004)

[13] Paul, M.J., Zhai, C.X., Girju, R. Summarizing contrastive viewpoints in opinionated text. *In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 66–76. Association for Computational Linguistics*, (2010)

[14] Paul, M., Girju, R. A two-dimensional topic-aspect model for discovering multi-faceted topics. *Proceedings of AAAI* (2010)

[15] Soboroff, I. Overview of the TREC 2004 novelty track. *The Thirteenth Text Retrieval Conference (TREC)* (2004)

[16] Androutsopoulos, Ion; Malakasiotis, P. A survey of paraphrasing and textual entailment methods. *Arxiv preprint arXiv:0912.3747* (2009)

[17] Cardie, C., Wiebe, J., Wilson, T., Litman, D. Combining low-level and summary representations of opinions for multi-perspective question answering. *In Working Notes-New Directions in Question Answering (AAAI Spring Symposium Series)*, (2003)

[18] Settles, B. Active Learning Literature Survey. *Computer Sciences Technical Report 1648, University of Wisconsin–Madison* (2009).

[19] Manning, C.D.; Schutze, H. *Foundations of statistical Natural Language Processing.* MIT Press (1999).

[20] Bing Liu. Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing, Second Edition*, (editors: N. Indurkhya and F. J. Damerau), (2010).

[21] Pang, Bo; Lee, Lillian. Opinion Mining and Sentiment Analysis. *Now Publishers Inc*. (2008)

[22] Andreas Thalhammer, Ioan Toma, Rakebul Hasan. Initial models for diversity-rich information. *RENDER Project Deliverable D3.1.1*. (2011)

[23] Arthur, D. and Vassilvitskii, S. *k*-means++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035*. (2007)

[24] Kim, S-M. and Hovy, E. Determining the sentiment of opinions. In Proceedings of COLING (2004).

[25] van Rijsbergen, C. J. Information Retrieval (2nd ed.). Butterworth. (1979)

Annex A

A.1 Paper: Expressing Opinion Diversity

Bizau, A**.,** Rusu, D**.,** and Mladenic, D. 2011. ***Expressing Opinion Diversity***. First International Workshop on Knowledge Diversity on the Web (DiversiWeb 2011), 20th World Wide Web Conference (WWW 2011). Hyderabad, India.

Link: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-762/paper2.pdf