entropy

MDPI

Article

Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis †

Łukasz Augustyniak [1,*], Piotr Szymański [1,2,*], Tomasz Kajdanowicz [1] and Włodzimierz Tuligłowicz [1]

Received: 10 August 2015; Accepted: 15 December 2015; Published: 25 December 2015
Academic Editors: J. A. Tenreiro Machado and Kevin H. Knuth

[1] Department of Computational Intelligence, Wrocław University of Technology, Wybrzeże Stanisława Wyspiańskiego 27, Wrocław 50-370, Poland; tomasz.kajdanowicz@pwr.edu.pl (T.K.); wlodzimierz.tuligłowicz@pwr.edu.pl (W.T.)
[2] Illimites Foundation, Gajowicka 64 lok. 1, Wrocław 53-422, Poland
* Correspondence: lukasz.augustyniak@pwr.edu.pl (L.A.); piotr.szymanski@pwr.edu.pl (P.S.)
† This paper is an extended version of our paper published in the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Beijing, China, 17–20 August 2014.

Abstract: We propose a novel method for counting sentiment orientation that outperforms supervised learning approaches in time and memory complexity and is not statistically significantly different from them in accuracy. Our method consists of a novel approach to generating unigram, bigram and trigram lexicons. The proposed method, called frequentiment, is based on calculating the frequency of features (words) in the document and averaging their impact on the sentiment score as opposed to documents that do not contain these features. Afterwards, we use ensemble classification to improve the overall accuracy of the method. What is important is that the frequentiment-based lexicons with sentiment threshold selection outperform other popular lexicons and some supervised learners, while being 3–5 times faster than the supervised approach. We compare 37 methods (lexicons, ensembles with lexicon's predictions as input and supervised learners) applied to 10 Amazon review data sets and provide the first statistical comparison of the sentiment annotation methods that include ensemble approaches. It is one of the most comprehensive comparisons of domain sentiment analysis in the literature.

Keywords: sentiment analysis; opinion mining; machine learning; ensemble classification; sentiment lexicon generation

1. Introduction

Sentiment analysis of texts means assigning a measure on how positive, neutral or negative the text is. It can be performed by experts, automatically or both, as different sentiment classifications can be treated as input to improve accuracy. In this paper we propose both a new lexicon generation scheme for automatic annotation and an ensemble based approach that provides competitive performance over slower and more complicated methods.

In the past when there was a need to annotate a text written in natural language, having enough resources, one would hire a group of human annotators, and employ them to read the texts and use their intelligence and knowledge to complete the task. To increase the accuracy of these results, different annotators would annotate a given text and then check how many annotations gave the same result. What lies behind such an approach is the intuition that if more people give the same response to the same text, the probability that the response is correct rises. On the other hand this approach is expensive, time consuming and may require sophisticated methods of selecting annotators to attain a real rise in accuracy.

With the rapid development of the internet and Web 2.0 era, user generated content became a reality. With the dawn of social media, the internet has become a treasure chest of information on people's thoughts. That was soon noticed by businesses and governments. Every day millions of people produce data in different form on various topics via Facebook, Twitter, blogs, forums *etc*. The scale, frequency and volume of generated data yields a natural need for automatic processing and classification. For this reason we are witnessing growing popularity of Sentiment Analysis and Natural Language Processing.

Nowadays sentiment analysis is used in many areas, e.g., predicting election outcomes [1], supplying organizations with information on their brands [2], summarizing products in reviews [3] or even predicting the stock market [4]. Every use case that depends on people's opinions can benefit strongly from automated sentiment analysis. There exists a big need for such analysis, although it must be performed in real-time with very high efficiency and high accuracy. The supervised learning approach provides the accuracy, the lexicon provides low time and memory complexity, but none of these approaches offers both characteristics. We wanted to develop a really fast and easy way to compute methods that provide accuracy similar to supervised learning methods.

Three main approaches to sentiment analysis are described in the literature: lexicon-based, supervised learning and unsupervised learning [5–7]. A well-illustrated introduction to different sentiment analysis approaches (as of 2011) can also be found in Bing Liu's tutorial from AAAI'2011 [8].

1.1. Lexicon-based Approach

The lexicon-based approach assumes that sentiment is related to the presence of certain words or phrases in the document: raw text in a processed structural representation. A lexicon is a set of features that have an assigned sentiment value. The sentiment of the document is annotated using these features from the lexicon that are (or are not) present in the document. Inferring the document's sentiment can be performed in different fashions: majority voting, averaging and thresholding or just plain counting.

One approach to generating lexicons is to create a set of expert selected sentiment markers (features) and then extend it using thesauri or more advanced language tools like WordNet; this approach is called dictionary-based. Bing Liu's lexicon [9] is a well-established example of such a lexicon. Liu extended his works with double propagation in [10]. In this paper the authors propose a method to assign polarities to newly discovered sentiment words in a domain.

Generating lexicons based on a corpus is another approach employed amongst others by Hatzivassiloglou *et al.* [11,12] who presented a method of generating a sentiment lexicon of adjectives based on a large corpus using log-likelihood approaches. Generating from a corpus can be performed using statistical and semantic methods. An example of the first is estimating sentiment based on frequency of occurrence in a large annotated corpus of texts [13]. Another statistical method uses the odds ratio (*i.e.*, the probability of being in a positive state and not in a negative divided by the contrary) employed in [14].

There exist also methods for automatic construction of a context-aware sentiment lexicon. Lu *et al.* [15] proposed an optimization framework that provides such context-aware lexicons. They underlined that sometimes in the same domain the same word may indicate different polarities with respect to different aspects. They provide an example of the word "large" that is negative for batteries while being positive for screens. Ding *et al.* [16] proposed a holistic lexicon-based approach to solving the problem by exploiting external evidence and linguistic conventions of natural language expressions. Their approach allows accounting for sentiment words that are context dependent, and which cause major difficulties for existing algorithms. Ding's approach tries to deal with special words, phrases and language constructs which impact on opinions based on their linguistic patterns. It also has an effective function for aggregating multiple conflicting opinion words in a sentence. They tried to deal with the problem of different sentiment orientation of the same work

in various contexts. For example, the word "long" in the following two sentences has completely different orientations, one positive and one negative: "*The battery of this camera lasts very long*" and "*This program takes a long time to run*". However, their method was tested on small datasets, and each of these datasets consisted of much lower than one thousand reviews.

Another automatic generation method of sentiment lexicons is presented by Mohammad *et al.* [17]. He created two state-of-the-art SVM classifiers, one to detect the sentiment of messages such as tweets and SMS (message-level task) and one to detect the sentiment of a term within a message (term-level task). They participated in the SemEval 2013 contest and among submissions from 44 teams in a competition, their submission came first in both tasks on tweets, obtaining an F-score of 69.02 in the message-level task and 88.93 in the term-level task. They implemented a variety of surface-form, semantic, and sentiment features. They also generated two large word–sentiment association lexicons, one from tweets with sentiment-word hashtags, and one from tweets with emoticons. Further information related to this approach is presented in Section 2.2.

We propose a new lexicon generation scheme that improves these approaches by assigning sentiment values to features based on both the frequency of their occurrence and the increase of how likely it is for a given feature to yield a given score (extending the basic log-likelihood approach) This method was named frequentiment (see Section 3.1). We calculate the document's frequentiment as an expected frequentiment over all features present in the document.

1.2. Supervised Learning Approach

A different and popular approach—supervised learning—is about learning from an available—already annotated—data set and making predictions for new cases [18]. To perform supervised learning one has to define a feature extractions method and apply it to data objects, one must have a training data set and choose a learning algorithm (classifier).

The data can be labeled manually by qualified human annotators, or sometimes labels can be derived from data itself, for example from the number of stars marked for a product review by the author of that review.

Pang and Lee [6] studied the problem of which text features give better results in sentiment analysis. They reported that unigrams (single words) along with part-of-speech tags and term frequency give most promising results.

Any existing supervised learning techniques can be used for sentiment classification. Researchers report high accuracy of classifiers such as Naive Bayes [19–24], Support Vector Machines (SVM) [19–23,25], and Decision Tree [21,26,27]. In most cases, SVM shows a slight improvement over Naive Bayes and Decision Tree classifiers, but remain much slower due to their computational complexity.

1.3. Ensemble Classification Approach

Lexicon-based methods in general are time-efficient and inaccurate in cases of sophisticated opinion texts. Also, lexicons may not scale well for specialized texts and are domain specific by nature. Medhat *et al.* [7] note that the dictionary based approach may fail to find opinion words with domain and context specific orientations; the corpus generation approach addresses this issue yet used alone may not be as effective as the dictionary-based approach. Supervised learning methods on the other hand are in general more accurate, but much slower than lexicon-based methods. Real world applications usually prefer an approach that provides a trade-off between these two conflicting optimization targets. Ensemble learning is a compromise approach between effectiveness and accuracy.

Whitehead [28] describes ensemble learning as a technique of increasing machine learning accuracy with a trade-off of increasing computation time so, best suited to in those domains where computational complexity is relatively unimportant compared to the best possible accuracy.

One of the first ensemble learning techniques was bootstrap aggregating (bagging). As described in [29], the bagging technique involves generating multiple versions of a predictor (using bootstrap replicates of the training set) and using them to form one aggregated predictor. Tests on real and simulated data sets show that bagging can give substantial gains in accuracy.

Another ensemble method is called boosting. Schapire [30] presented its basic idea as consisting of three steps: (1) performing an iterative search to locate the regions/examples that are more difficult to predict, (2) rewarding accurate predictions on those regions in each iteration, (3) combining the rules from each iteration. He also presented a version of a boosting algorithm, called AdaBoost (Adaptive Boosting), which solved many of the practical difficulties of its predecessor.

Various other ensemble algorithms exist and differ usually in how they answer the three basic questions presented in [31]: (1) How are subsets of the training data chosen for each individual learner? (2) What types of learners are used to form the ensemble? (3) How are classifications made by the different individual learners combined to form the final prediction?

A very limited evaluation by Whitehead *et al.* in [14] shows that bagging can provide an increase of accuracy up to 3 percentage points, while boosting may suffer from overfitting, yet the results can hardly be considered statistically significant. We employ a variety of ensemble approaches to provide statistically significant results.

1.4. Our Contribution

In this paper we would like to present the continuation of our work presented in [27], where we have used sentiment lexicons as first stage classifiers and then employed a decision tree as a fusion classifier, which learned based on the output of the lexicons. This approach, as predicted, did not increase significantly the overall accuracy, but did decrease the computation time approximately 200 times and has lower memory complexity.

The contributions of this paper are three-fold. Firstly, a new method for lexicon generation based on margin frequency and the likelihood approach was presented. Secondly, we overcome the dichotomy between general-purpose and domain-specific sentiment lexicons by employing a wide range of ensemble approaches that assign sentiment based on input from multiple lexicons, thus reconciling both domain-specific and general purpose knowledge. Finally, we strive to provide the first statistically significant results concerning ensemble-approach performance and compare them to the well-established supervised learning baseline.

Compared with previous proposal in our conference paper the lexicon generation is extended and a more complex analysis of the lexicons is presented. Ensemble classification is extended with additional classifiers (Random Forests, different version of Naive Bayes, Linear SVC, Logistic Regression, Extra Tree Classifier and AdaBoost). The baseline supervised learning approach was extended with new classifiers for comparison purposes. The whole experiment was conducted on larger datasets from various domains (10 different domain from Amazon Dataset SNAP [32]). This paper presents a wide comparison and analysis of sentiment task for several approaches, lexicons and product domains.

In this paper we state two hypotheses upon which we construct a new approach to sentiment analysis:

(1) Exploitation of the impact of unigram/bigram/trigram regarding both frequency and likelihood on a corpus upon its sentiment yields better results than expert annotation.
(2) Employment of ensemble techniques on multiple lexicon outputs for learning of sentiment annotation schemes yields results comparable to supervised methods but is more efficient.

We evaluate these hypotheses by comparing proposed approaches to established methods on 10 domains, with 10 cross-validations, and 42,000 reviews per each. The hypotheses are tested using the Friedman-Iman-Davenport non-parametric rank test with the Nemenyi post-hoc procedure.

The rest of this paper is organized as follows: In Section 2 we describe the experiment design. In Section 3 we explain our proposed methods. Section 4 is about baselines used to compare the method with. In Section 5 we present and interpret the results obtained which we further relate to the state of the literature in Section 6. Finally, in Section 7 we conclude and present ideas for future work.

2. Experiment Design

In this section the experimental scenario—dataset, text pre-processing, cross-validation division are presented. The experiment consists of 29 sentiment classification methods that represent three types of approaches:

(1) Lexicon-based annotation.
(2) Frequentiment-based lexicon generation.
(3) Ensemble classifiers with lexicon sentiment prediction as input features.
(4) Supervised learning classifiers used for comparison purposes.

2.1. Dataset and Data Preparation

These methods were evaluated on the Amazon Reviews data set published by SNAP [32]. The following 10 domains of reviews were chosen for the experiment (the total count of reviews is presented below):

- Automotive product reviews (188,728 reviews)
- Book reviews (12,886,488 reviews)
- Clothing reviews (581,933 reviews)
- Electronics product reviews (1,241,778 reviews)
- Health product reviews (428,781 reviews)
- Movie TV reviews (7,850,072 reviews)
- Music reviews (6,396,350 reviews)
- Sports Outdoor product reviews (510,991 reviews)
- Toy Game reviews (435,996 reviews)
- Video Game reviews (463,669 reviews)

Each review consists of the Amazon user opinion (text) and its star score (1–5 scale), where 1 is the worst score and 5 is the best. The review data set was cleaned up from its raw form. All the HTML tags and entities were removed or converted to textual representations using the HTML parser in python library BeautifulSoup4 [33]. Next the unicode review texts were decoded to ASCII using the unidecode [34] python library. In addition, all punctuation marks and numbers were removed.

Each of the data sets was divided into a training and test set in 10 cross-validations. For tractability, especially with supervised learners, the training data set consisted of 12,000 randomly drawn reviews. Reviews were selected evenly per sentiment, *i.e.*, the training set included 2000 reviews with 1, 2, 4 and 5 stars each and 4000 labeled with 3 stars. We have thus obtained a balanced set of 4000 positive, negative and neutral reviews each. The test data set consisted of 30,000 evenly distributed across sentiment labels (distributed analogously to the training set).

In order to check the accuracy of the proposed methods, the ground truth sentiment was extracted from ratings expressed with stars. Ratings were mapped to the text classes ”positive”, ”neutral” and ”negative”, using 1 and 2 stars, 3 stars, 4 and 5 stars respectively, see Table 1.

Table 1. Star rating mapping to sentiment classes.

Star Score	Sentiment Class
★	Negative
★★	Negative
★★★	Neutral
★★★★	Positive
★★★★★	Positive

2.2. Sentiment Lexicons

Several lexicons, both fixed and dynamically-generated were used in the experiments. Fixed lexicons with exemplary content words were presented in Table 2. Exemplary lexicons, generated by Augustyniak *et al.* [27] and SO-PMI/LSA generated based on Turney *et al.* [35], Bing Liu's Opinion Lexicon [9], AFINN Lexicons [36], list of positive/negative words from www.enchantedlearning.com, MPAA lexicon [37] and lexicons from NRC Canada group [17,38,39] are all polarized lexicons.

Table 2. Examples of sentiment lexicons.

Lexicon	Positive Words	Negative Words
Simplest (SM)	good	bad
Simple List (SL)	good, awesome, great, fantastic, wonderful	bad, terrible, worst, sucks, awful, dumb
Simple List Plus (SL+)	good, awesome, great, fantastic, wonderful, best, love, excellent	bad, terrible, worst, sucks, awful, dumb, waist, boring, worse
Past and Future (PF)	will, has, must, is	was, would, had, were
Past and Future Plus (PF+)	will, has, must, is, good, awesome, great, fantastic, wonderful, best, love, excellent	was, would, had, were, bad, terrible, worst, sucks, awful, dumb, waist, boring, worse
Bing Liu	2006 words	4783 words
AFINN-96	516 words	965 words
AFINN-111	878 words	1599 words
enchantedlearning.com	266 words	225 words
MPAA	2721 words	4915 words
NRC Emotion	2312 words	3324 words

The polarized lexicons, that we use are lists of words w with an assigned numeric value 1 for positive, 0 for neutral and -1 for negative sentiments. The polarity of the document is calculated based on detecting occurrences of sentiment words w from the lexicon l in that document $d = \{w_1, w_2, ..., w_k\}$.

Let:

$$pos(l,d) = \#\text{ of positive words from } l \text{ that occur in } d \tag{1}$$

$$neg(l,d) = \#\text{ of negative words from } l \text{ that occur in } d \tag{2}$$

$$sum(l,d) = pos(l,d) - neg(l,d) \tag{3}$$

The sentiment orientation $s_l(d)$ of a document d under a polarized lexicon l is assigned using the following formula:

$$s_l(d) = \begin{cases} 1 & if\ sum(l,d) > 0 \\ -1 & if\ sum(l,d) < 0 \\ 0 & otherwise \end{cases} \tag{4}$$

3. Proposed Methods

In this subsection our proposed method is described. It consists of two steps:

(1) Frequentiment lexicon generation (unigram, bigram and trigram lexicons are generated).
(2) Ensemble classification (fusion classifier) step.

3.1. Novel Frequentiment-based Lexicon Generation

We propose a novel method for generating lexicons from corpora by evaluating features present in corpus documents. The unigrams, bigrams and trigrams are taken as features. Some pre-processing steps were taken. Unigrams shorter than 4 characters were omitted, but such short words were included for bigram and trigram lexicon generation. Bigrams and trigrams were extracted from words occurring consecutively within one sentence. This way we take into account the relational structure present in consecutive word co-occurrence instead of just using single-word features as is the standard in many lexicon generation techniques. Any ngram that occurred in less than 1% of reviews in the training set were removed. Each ngram is counted only once per document. Words and sentences are tokenized using NLTK's `tokenize` module.

Then for each of the ngram (separately for each domain and cross-validation set) a frequentiment score was calculated:

$$fqmt(f) = \sum_{s \in scores} s \cdot \frac{P(\text{review has score } s | \text{review has feature } f)}{P(\text{review has score } s)} \tag{5}$$

The frequentiment measure captures how the presence of an ngram increases the likelihood of a certain star score in review, averaged over all scores. It is calculated per feature and as it expresses the average of probability ratios (it is not normalized). A document's frequentiment is obtained as a mean frequentiment of the document's features, averaged according to features present in the document. To make it more clear let us define a helper random variable over the universum of features and documents:

$$X(f,d) = \begin{cases} fqmt(f) & \text{if } f \in d \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

In the case of unigrams, unique unigrams are extracted from the document. Bigrams and trigrams are extracted per sentence, and then unique bigrams and trigrams per document are selected. For a given generated frequentiment lexicon l and a given document d the frequentiment is thus calculated as:

$$fqmt(d) = E[X|d] = \sum_{f \in l} X(f,d) = \sum_{f \in d} fqmt(f) \tag{7}$$

As the lexicon is not a sentiment polarity lexicon due to lack of normalization—it is a measure of likelihood increase not probability—a threshold needs to be selected for converting a document's d frequentiment score to sentiment polarity using a selected threshold:

$$s_{\text{frequentiment}}(d) = \begin{cases} -1 & \text{if } fqmt(d) \leq -t \\ 0 & \text{if } fqmt(d) \in (-t,t) \\ 1 & \text{if } fqmt(d) \geq t \end{cases} \tag{8}$$

We estimated the lexicon's best parameter by evaluating the lexicon's F-measure for parameters $t \in [0,10]$ with a step of 0.1 on training sets per cross-validation per domain. For experimental purposes we selected the best parameter value averaged over all cross-validation folds for each of the domains.

3.2. Ensembles of Weak Classifiers

The second part of our proposed methods concerns the lexicon ensemble approach. It consists of two stages—building the relevant input space for ensemble classification, and learning a fusion classifier based on mentioned input space. This part of our method uses a variety of models (lexicons in this experiment) whose predictions are taken as input to a new model that learns how to combine the predictions into an overall prediction. We built a sentiment polarity matrix $S(\mathcal{L}, \mathcal{D})$ using predictions from various sentiment lexicons. Sentiment orientation was obtained for every document $d \in \mathcal{D} = \{d_1, \ldots, d_n\}$ and every lexicon $l \in \mathcal{L} = \{l_1, \ldots, l_n\}$. We denoted the sentiment polarity of a document d using lexicon l as $s_l(d)$ regardless. The sentiment polarity matrix is defined as follows:

$$\mathcal{S}(\mathcal{L}, \mathcal{D}) = \begin{pmatrix} s_{l_1}(d_1) & s_{l_1}(d_2) & \cdots & s_{l_1}(d_n) \\ s_{l_2}(d_1) & s_{l_2}(d_2) & \cdots & s_{l_2}(d_n) \\ \vdots & \vdots & \ddots & \vdots \\ s_{l_n}(d_1) & s_{l_n}(d_2) & \cdots & s_{l_m}(d_n) \end{pmatrix} \tag{9}$$

The experiment was conducted with different versions of Naive Bayes classifiers (Gaussian, Multinomial and Bernoulli), Linear SVC, Logistic Regression, Extra Tree Classifier and AdaBoost learner. The whole ensemble classification is visually presented in Figure 1.

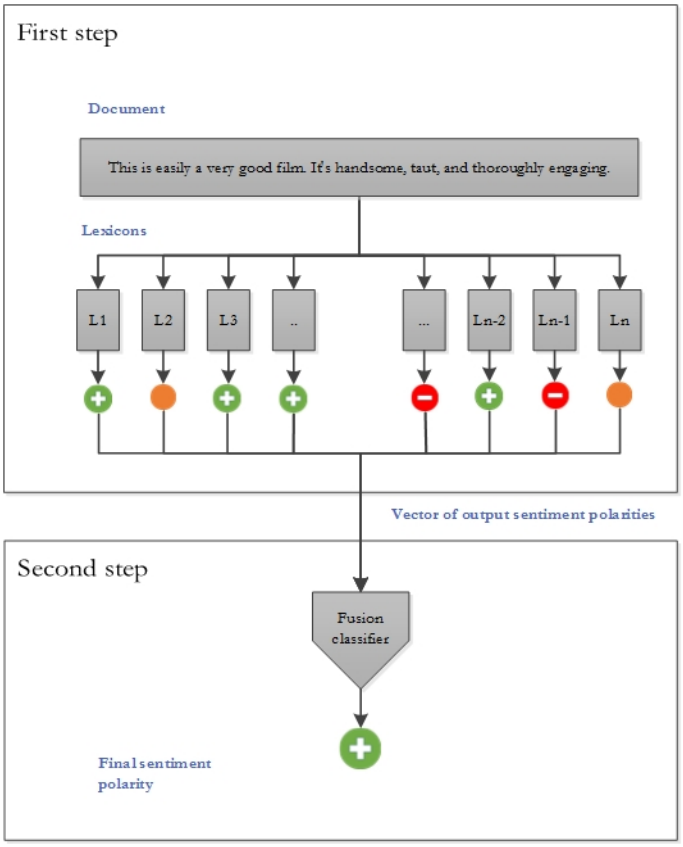


Figure 1. The concept of the proposed ensemble classification.

4. Baselines

4.1. Semantic Orientation Lexicons

To compare our approach with other lexicon generation approaches we turn to the semantic orientation lexicon generation method. It is based on an assumption that sentiment orientation of

a word is the same as that of the words that co-occur with that word. This approach presented by Turney *et al.* [40] defines a basic initial set of positive and negative oriented words, a kernel so to speak, which is then extended using two methods of mining and assessing co-occurrence of words in the text by PMI (*Pointwise Mutual Information*).

The method starts with two seed sets of opposing words:

$$Pwords = \{good, nice, excellent, positive, fortunate, correct, andsuperior\}$$

$$Nwords = \{bad, nasty, poor, negative, unfortunate, wrong, inferior\}$$

PMI (*Pointwise Mutual Information*) is defined as below:

$$PMI(word_1, word_2) = log_2\Big(\frac{p(word_1 \& word_2)}{p(word_1)p(word_2)}\Big) \tag{10}$$

We count the probabilities $p(word_1 \& word_2)$ using empirical distributions on a given training set. It is understood as probability of $word_1$ and $word_2$ occurring in the same document.

For each word from the corpus we count SO-PMI (Semantic Orientation PMI) as follows:

$$\mathrm{SO}_{\mathrm{PMI}}(word) = \sum_{pword \in Pwords} PMI(word, pword) - \sum_{nword \in Nwords} PMI(word, nword) \tag{11}$$

In addition, we used the SO-LSA method, it is based on Latent semantic analysis (LSA) approach. We calculated the TF-IDF matrix for a given training set and used cosine distance between angles of corresponding word vectors, which is denoted as $LSA(word1, word2)$. LSA uses the Singular Value Decomposition (SVD) to lower the number of considered features, *i.e.*, the dimensions of the TF-IDF matrix. We conducted the SVD with $k = 150$ dimensions, the optimal value from Turney's experiment. Thus in a similar fashion to SO-PMI we define SO-LSA as:

$$\mathrm{SO}_{\mathrm{LSA}}(word) = \sum_{pword \in Pwords} LSA(word, pword) - \sum_{nword \in Nwords} LSA(word, nword) \tag{12}$$

For a given document we calculated the expected orientation using the method over all words present in the document. For the SO-PMI mean value for all sentiment orientations of words was counted.

$$\mathrm{SO}_{\mathrm{PMI}}(d) = \sum_{word \in d} \mathrm{SO}_{\mathrm{PMI}}(w)$$

For the SO-LSA according to the paper [40] the average sentiment orientation was used.

$$\mathrm{SO}_{\mathrm{LSA}}(d) = \frac{1}{|d|} \sum_{word \in d} \mathrm{SO}_{\mathrm{LSA}}(w)$$

As the lexicon is not a sentiment polarity lexicon due to lack of normalization—a threshold needs to be selected for converting a document d's SO score to sentiment polarity using a selected threshold:

$$s_{\mathrm{PMI/LSA}}(d) = \begin{cases} -1 & \text{if } \mathrm{SO}_{\mathrm{PMI/LSA}}(d) \leq -t \\ 0 & \text{if } \mathrm{SO}_{\mathrm{PMI/LSA}}(d) \in (-t, t) \\ 1 & \text{if } \mathrm{SO}_{\mathrm{PMI/LSA}}(d) \geq t \end{cases}$$

After parameter estimation we have selected a $t = 1.1$ for SO-PMI method and $t = 0.1$ for SO-LSA approach. It is average of the best thresholds across all domains.

4.2. Supervised Learning Baseline

In the three class classification the obvious baseline for results would be $33.(3)\%$ and the computational complexity would be $O(1)$. For the evaluation of our method we would like to use a well-known and widely used baseline that has been by Pang *et al.* in [6] and several other authors. The chosen baseline method as features uses single words (unigrams) occurring in the test set that "survived" the pre-processing stage. Words that occurred in the test set, but were not found in the feature set are ignored. An input document is turned into an input vector. A Bag-of-Words model was used in this experiment. It treats each document as a vector, where length of the vector is the size of a vocabulary (a list of all unique words derived from the whole corpus). Each element (word) in the vector represents the number of times that the word appears in the document. For example, the phrase *I liked it, liked it very much* might be encoded as [0, ..., 1, 2, 2, 1, 1, ..., 0]. The 0 represents here words which do not appear in the document. The matrix based on such a document's vectors is treated as a feature space for supervised learning algorithms.

In that case the size of feature space depends on the size and variety of words in the training data. The feature space for large corpora may be outrageously big, and the input vectors are very sparse, which is really common. This approach is memory and computationally demanding. We obtained feature spaces as follows (average number of features for 10 folds of cross-validation):

- Automotive: 36,789,
- Books: 82,541,
- Clothing & Accessories: 22,872,
- Electronics: 50,758,
- Health: 40,330,
- Movies & TV: 81,380,
- Music: 79,969,
- Sports & Outdoors: 40,956,
- Toys & Games: 40,253,
- Video Games: 63,471.

Any classifier can be used in this method, because our representation of the Bag-of-Words consists of only numerical vectors. Classifiers such as BernoulliNB, DecisionTreeClassifier, LinearSVC, LogisticRegression, MultinomialNB from Python scikit-learn (scikit-learn.org) library were used in the experiment.

We chose a supervised learning approach as a baseline, because we wanted to have a well-known and world-wide used method for comparison purposes. This method is trained and tested for each separate domain, hence it could be treated as a baseline for our domain dependent ensemble method. It is worth mentioning that a supervised learning approach with unigrams and bigrams as features and a Logistic Regression classifier (the best classifier in our experiments, see Section 5) achieves approximately 65% of F-measure for SemEval 2013 Twitter Data [41] that is really close to the best score 69% of NRC-Canada in [17]. It should be underlined that our experiment was conducted on the review dataset (texts longer than tweets). The characteristic of long texts is quite different from tweets, hence we are not sure if Twitter-based lexicons are applicable in this situation.

For the purposes of obtaining the first comparison of ensemble-based approaches that yields statistical significance we have considered the performance of 14 lexicons and 5 supervised learners used as a state-of-the-art baseline.

5. Results

The results from the experiment described are presented in Table 3 and Figures 2–4. The performance was evaluated by the accuracy, recall, precision and F-measure for each method and domain. However, we present only F-measures above because of space limitations in this paper.

The last row in the Table presents the average for F-measure across each domain. In addition, the comparison of efficiency using consumed memory and execution time was conducted and described.

Table 3. Results for all methods - F-measure.

	Method	Auto	Books	C&A	Elect	Health	M&TV	Mus	SP	T&G	VG
Lexicon-based approach	SM	0.244	0.227	0.249	0.244	0.245	0.230	0.228	0.245	0.230	0.237
	SL	0.335	0.333	0.341	0.349	0.342	0.382	0.357	0.343	0.344	0.367
	Emotion	0.342	0.355	0.360	0.354	0.347	0.366	0.352	0.358	0.350	0.359
	PF	0.352	0.365	0.361	0.348	0.362	0.368	0.340	0.362	0.379	0.357
	SL+	0.351	0.364	0.385	0.364	0.366	0.398	0.362	0.366	0.376	0.395
	AF-111	0.368	0.346	0.364	0.376	0.358	0.370	0.350	0.368	0.359	0.368
	PF+	0.366	0.375	0.411	0.360	0.376	0.389	0.335	0.381	0.387	0.370
	trigr.	0.348	0.395	0.361	0.386	0.380	0.390	0.353	0.366	0.392	0.388
	AF-96	0.390	0.364	0.391	0.401	0.387	0.385	0.371	0.398	0.395	0.388
	EN	0.419	0.389	0.400	0.406	0.411	0.394	0.391	0.410	0.411	0.394
	MPAA	0.386	0.380	0.406	0.392	0.381	0.391	0.374	0.403	0.404	0.383
	BL	0.411	0.387	0.421	0.414	0.410	0.406	0.407	0.429	0.439	0.404
	bigr.	0.440	0.461	0.496	0.498	0.503	0.457	0.370	0.472	0.500	0.495
	unigr.	**0.500**	**0.505**	**0.530**	**0.508**	**0.505**	**0.512**	**0.435**	**0.514**	**0.511**	**0.499**
Supervised learn	DT	0.600	0.478	0.770	0.484	0.521	0.486	0.474	0.569	0.528	0.491
	BNB	0.631	0.542	0.737	0.568	0.591	0.542	0.506	0.617	0.606	0.546
	LinSVC	0.647	0.552	0.799	0.549	0.577	0.557	0.554	0.617	0.590	0.554
	MNB	0.631	0.564	0.740	0.571	0.587	0.567	0.573	0.618	0.614	0.564
	LogR	**0.664**	**0.584**	**0.800**	**0.581**	**0.604**	**0.587**	**0.587**	**0.640**	**0.621**	**0.586**
Lexicon-based ensemble	NMB	0.360	0.411	0.380	0.374	0.357	0.401	0.419	0.371	0.395	0.403
	BNB	0.424	0.402	0.409	0.410	0.416	0.398	0.427	0.424	0.451	0.403
	LogR	0.441	0.448	0.461	0.451	0.428	0.453	0.441	0.444	0.463	0.446
	DT	0.491	0.460	0.562	0.459	0.459	0.465	0.456	0.484	0.490	0.457
	GNB	0.460	0.477	0.490	0.494	0.472	0.482	0.468	0.479	0.494	0.489
	ET	0.495	0.461	0.568	0.460	0.461	0.467	0.457	0.487	0.493	0.459
	RF	0.513	0.482	**0.596**	0.479	0.482	0.488	0.473	0.508	0.512	0.480
	AB	**0.521**	**0.525**	0.540	**0.536**	**0.529**	**0.535**	**0.510**	**0.528**	**0.552**	**0.530**
	AVG	0.449	0.431	0.494	0.438	0.439	0.439	0.421	0.452	0.455	0.437

Domains: Automotive, Books, Clothing & Accessories, Electronics, Health, Movies & TV, Music, Sports & Outdoors, Toys & Games, Video Games. **Methods**: lexicons as described in Table 2 with extension of unigrams, bigrams, trigrams, NRC Emotion and MPAA. **Classifiers**: **DT**—Decision Tree, **BNB**—Bernoulli Naive Bayes, **LinSVC**—Linear SVC, **MNB**—Mulinomial Naive Bayes, **LogR**—Logistic Regression, **AB**—AdaBoost, **ET**—Extra Tree Classifier, **RF**—Random Forest and **GNB**—Gaussian Naive Bayes. **AVG**—average F-measure score for each domain.

There is a significant difference in the results for different domains, indicating that some of the domains are easier to analyze than others. In particular, the Clothes and Accessories domain gets much better results than others (0.8 for Logistic Regression) and it is higher than the average value of F-measure which is 0.494 (0.505 ± 0.06 if omitting outliers).

The results obtained were described in the following order:

(1) Parameter estimation for the frequentiment lexicon generation.
(2) Lexicon performance, frequentiment-generated versus state-of-the-art lexicons.
(3) Lexicon and lexicon-based ensemble methods compared to supervised learners.

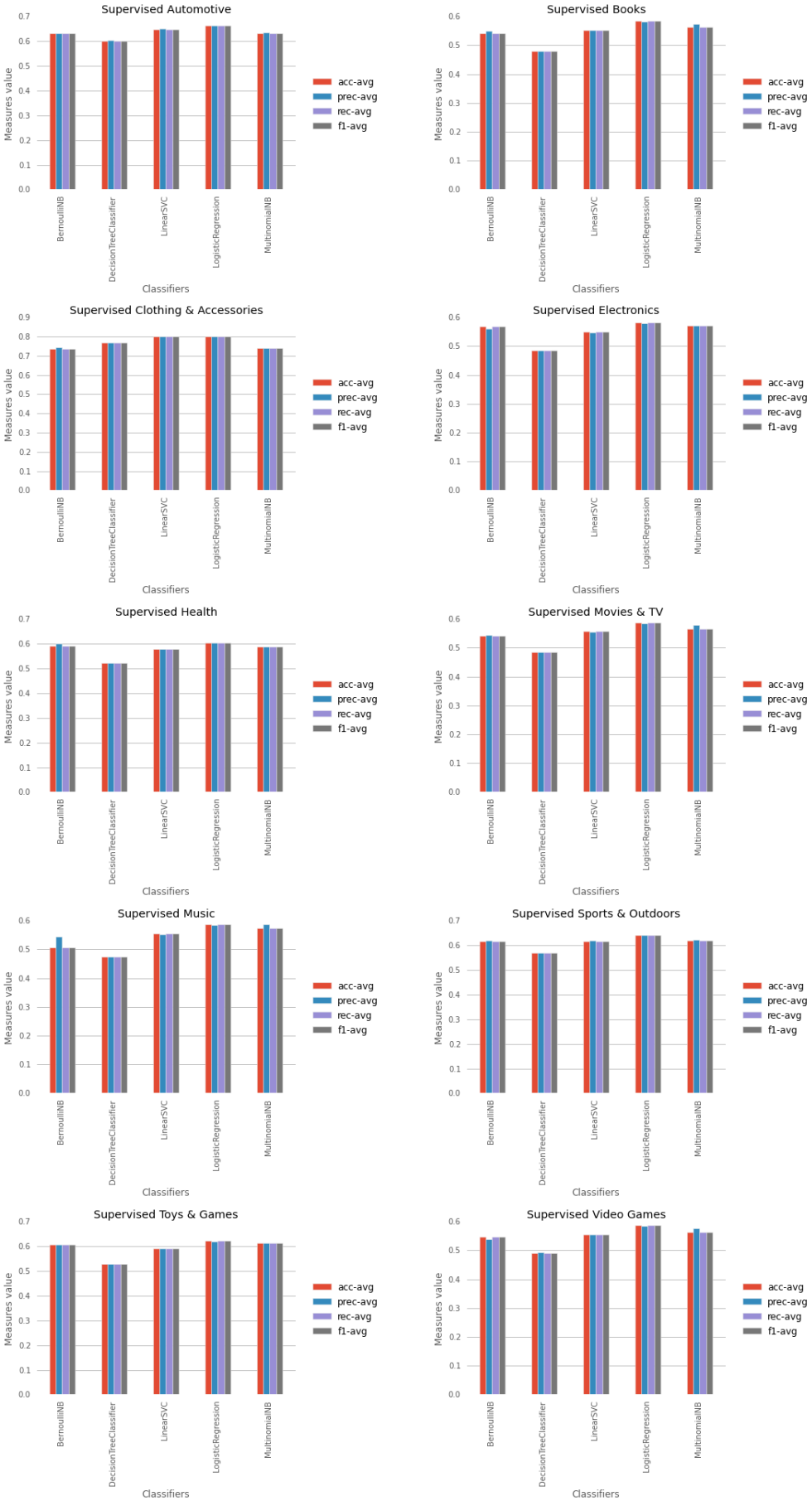


Figure 2. Supervised learning baseline–various domains.

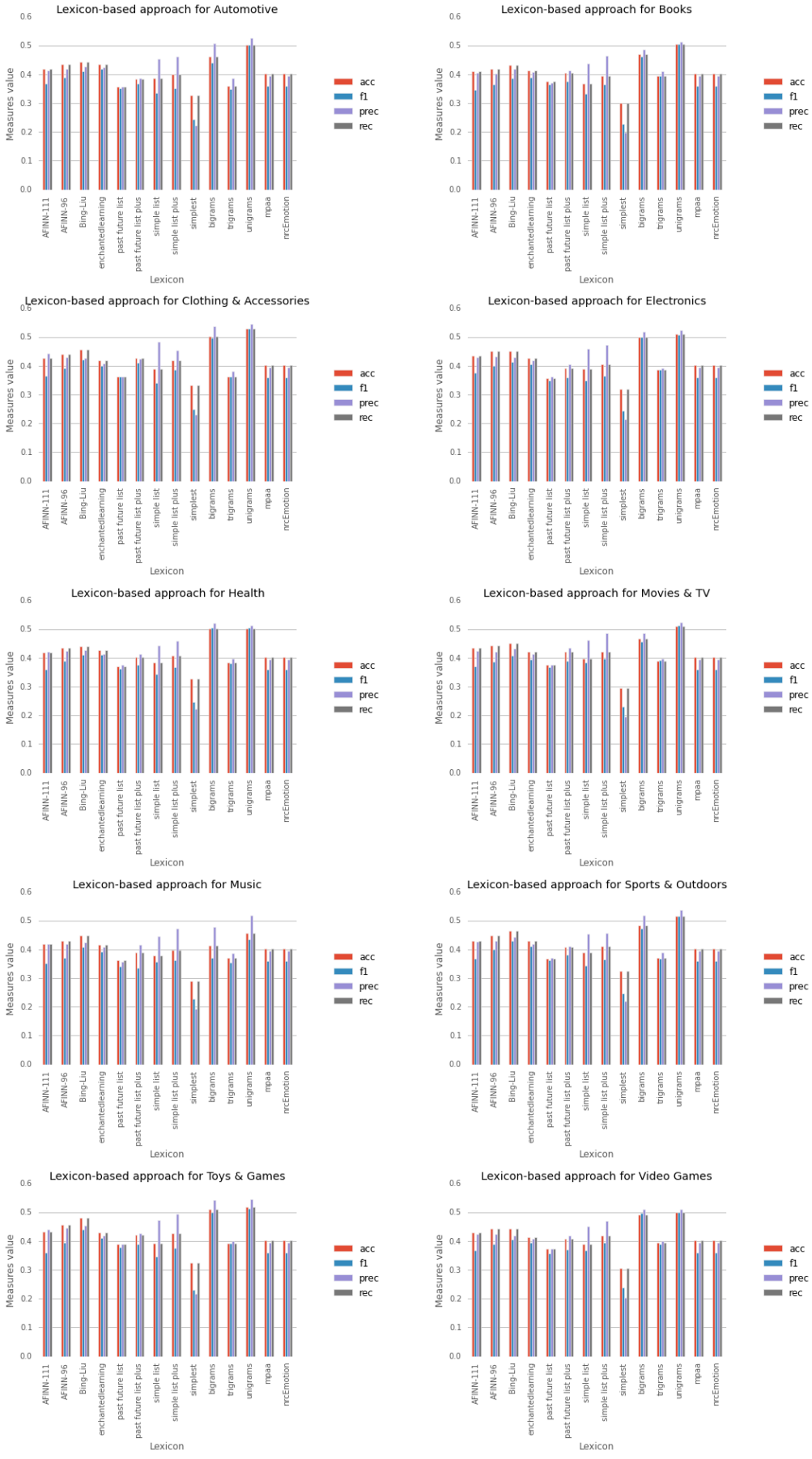


Figure 3. Lexicon-based learning—various domains.

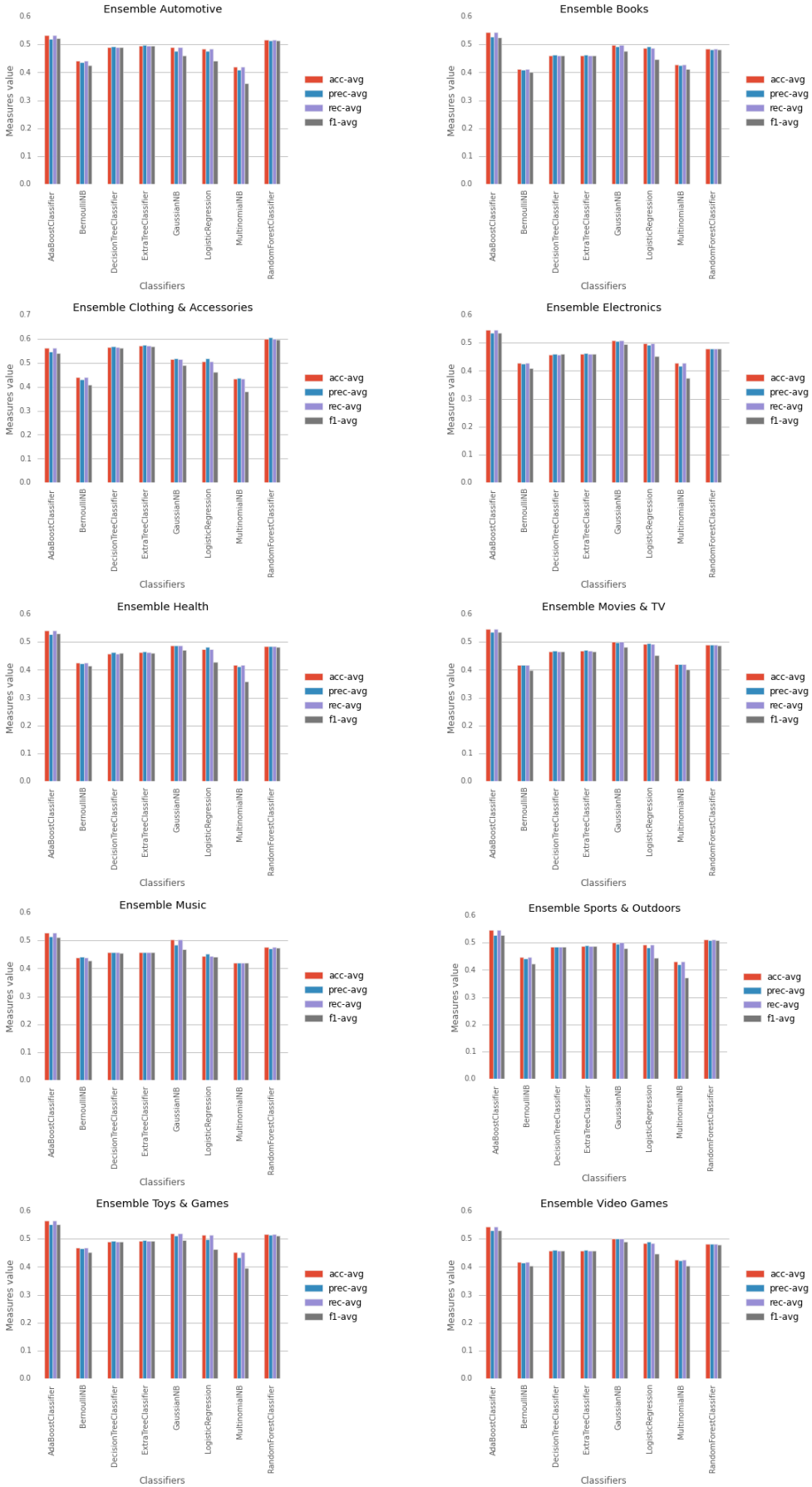


Figure 4. Ensemble classification with lexicons—various domains.

5.1. Parameter Estimation for Frequentiment Lexicon Generation

Our first results concern selection of the threshold for lexicon-based approach. We obtained F-measures per threshold for unigrams, bigrams and trigrams. Firstly, we evaluate the best average results, as averaged over 10 cross-validations for every problem. For unigrams such as the best average F-measure was maximized with thresholds 1.2–1.7 and the average best threshold over all problems of 1.43. For bigrams it was respectively 1.1–2.0 with best average threshold of 1.54. For trigrams it was 0.1–0.4, where 0.4 was a strong outlier, and the average best threshold was 0.19. Table 4 presents thresholds across all domains.

Table 4. Achieved frequentiment lexicon results at best thresholds average and standard deviation.

	Unigrams	**Bigrams**	**Trigrams**
Automotive	0.500 ± 0.008	0.440 ± 0.015	0.348 ± 0.005
Books	0.505 ± 0.008	0.461 ± 0.016	0.395 ± 0.005
Clothing & Accessories	0.530 ± 0.005	0.496 ± 0.007	0.361 ± 0.006
Electronics	0.508 ± 0.008	0.498 ± 0.010	0.386 ± 0.004
Health	0.505 ± 0.004	0.503 ± 0.009	0.380 ± 0.003
Movies & TV	0.512 ± 0.005	0.457 ± 0.019	0.390 ± 0.004
Music	0.435 ± 0.015	0.370 ± 0.018	0.353 ± 0.003
Sports & Outdoors	0.514 ± 0.006	0.472 ± 0.014	0.366 ± 0.003
Toys & Games	0.511 ± 0.008	0.500 ± 0.010	0.392 ± 0.004
Video Games	0.499 ± 0.007	0.495 ± 0.005	0.388 ± 0.007

The frequentiment lexicon F-measure as a function of the thresholding parameter can be observed in Figure 5. For all of the features, both unigram, bigrams and trigrams, this function is unimodal, with a maximum around the aforementioned values. Intuitively, the more extreme the frequentiment (both positive or negative) the larger the total average impact of the frequentiments of features. Thus a very high frequentiment of 7 should indicate an extremely positive review. If this were the case, the function should be a monotonically rising curve. But it is not possible in practice, as there is a limit on how many impactful features can be found in any document due to natural characteristics of the language.

A well-written review cannot consist of just the impactful words and even if some of the reviews did, they would not span 1% of the corpus as required in the experimental conditions. Regarding these functions, with consistent unimodal behaviour across all the data sets, we observer that the optimal threshold is a compromise between the most impactful words and frequency of how often they can appear in a document. The average case of 1.43 frequentiment-marked likelihood increase of a polarized score does seem sensible. If the document contains words that after averaging increase the likelihood of it being positive or negative 1.43 times, it should be a good indicator.

On the other hand if we require all positive or negative documents to provide a 6-times increase of a polarized sentiment's likelihood to mark them as positive or negative, we end up discarding a lot of legitimately positive or negative documents, and thus in many case a threshold of 6 causes the lexicons to perform poorer than the 0.33 baseline. A similar argument holds for bigrams.

The different case for trigrams stems from a small list of trigrams in the lexicons. It is related to the nature of observed trigrams which are described in the next subsection and can be observed in examples in Figure 6. There is rarely more than one trigram present per sentence and rarely more than a few per document. All in all the number of trigrams that are found in the lexicons is 7–11 times smaller than the number of unigrams or bigrams extracted from the same data set.

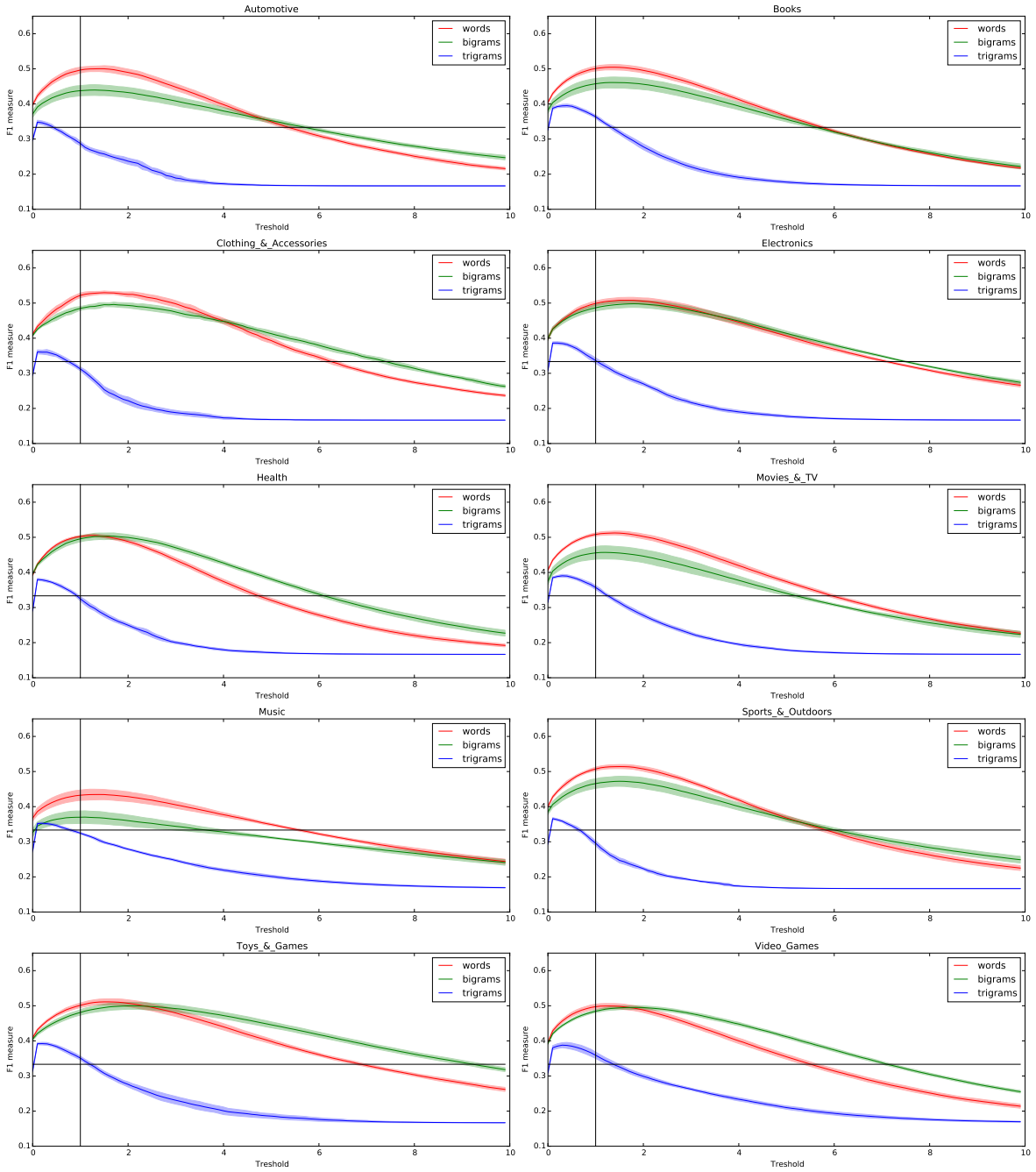


Figure 5. F-measure of unigram, bigram and trigram lexicons per domain as changed by neutrality threshold selection.

Figure 6. Exemplary positive and negative frequentiment ngrams (unigrams, bigrams and trigrams) for each domain. The size of the feature is proportional to its frequentiment.

5.2. Frequentiment Lexicon Performance and Characteristics

Achieved best scores for each data set are presented in Table 3. Results among 10 cross-validations deviate only by 1–2 percentage points for unigrams and bigrams, and by less than 1 percentage point for trigrams. In all cases frequentiment data scored higher than the 0.3 baseline. F-measures achieved by unigrams are close to 0.5 apart from the Music data set which scored the lowest 0.435, and span from 0.499–0.514 if outliers removed. The highest outlier for unigrams is Clothing & Accessories, with 0.53. Bigrams results span from a 0.37 outlier on Music data set to 0.503 on Health reviews while with outliers removed the scores concentrate in the interval of 0.44–0.5. The trigram approach yields to F-measure in the interval of 0.346 to 0.376 which is remarkable for such a small lexicon.

The size of obtained lexicons varies with **356.9–660.9** unigrams (averages per data set over cross-validations), **531.8–935.4** bigrams and **52.5–171.3** trigrams. The averages and standard deviations of the frequentiment lexicon's sizes are presented in Table 5. In the following paragraphs we provide a list of the top 1% positive and negative markers per data set.

Table 5. Frequentiment lexicon sizes.

	Unigrams	Bigrams	Trigrams
Automotive	372.1 ± 4.3	536.2 ± 6.7	52.5 ± 1.7
Books	641.7 ± 10.7	877.3 ± 14.2	140.5 ± 5.0
Clothing & Accessories	290.6 ± 4.0	531.8 ± 6.6	63.8 ± 2.4
Electronics	514.2 ± 7.7	727.3 ± 11.8	95.1 ± 4.3
Health	356.9 ± 5.8	563.0 ± 5.7	68.8 ± 1.4
Movies & TV	660.9 ± 3.6	896.7 ± 7.2	137.6 ± 2.7
Music	548.6 ± 5.1	817.0 ± 14.1	121.3 ± 4.3
Sports & Outdoors	362.2 ± 5.2	552.0 ± 6.9	62.0 ± 3.3
Toys & Games	379.4 ± 5.2	682.7 ± 7.5	98.4 ± 2.7
Video Games	549.4 ± 9.9	935.4 ± 8.5	171.3 ± 4.0

5.2.1. Analysis of the Most Prominent Lexicon Features

As shown in Figure 6, it can be seen that the frequentiment measure captures both the general, dictionary-approach words that are consistent in sentiment across problems and the problem specific words.

In terms of unigrams, it has detected the generally positive words such as *excellent* (average frequentiment of 2.23 ± 0.41), *superb* (2.41 ± 0.23), *amazing* (2.08 ± 0.28) or *highly* (2.8 ± 0.7); and negative ones such as *waste* (-3.95 ± 0.28), *return* (-2.07 ± 1.08) or *refund* (-3.92 ± 0.24). On the other hand it came across relevant problem-specific terms, such as: *addictive* (2.39 frequentiment, found only in Video Games), *comfortable* (1.53 ± 0.38 present only in Automotive, Electronics, Sports, Health and Clothing), *garbage* (-3.53, only in Music), *repair* (-1.62 ± 1.06 in Automotive and Electronics), *journey* ($1.61\pm$, Books), *cheaply* (-2.42, Toys), *ripped* (-1.98, Clothes) and many more. A detailed analysis with an English language expert might bring up more interesting characteristics, yet there are too many to describe here in depth.

In terms of bigrams, a large set is constituted by negations, *i.e.*, bigrams of the form "not X" where X would be a word feature. Apart from the negations, interesting findings of general markers include *piece of* (one can only imagine of what, present in 8 domains, -1.64 ± 0.7), *should have* (all domains, -1.35 ± 0.26), *supposed to* (9 domains, -1.24 ± 0.37). The positive markers include strong unigrams with a generally neutral object or verb such as *recommend it* (1.6 ± 0.22, 7 domains) or *would recommend* (1.84 ± 0.44, 8), or adjective gradation *the best* (1.9 ± 0.34, 10), *very good* (1.29 ± 0.47, 10). Alongside the general ones we can easily see domain specific ones: *work pants* (3.55, Clothes), *she loves* (2.31, Toys), *my only* (1.92 ± 0.29, Clothes, Electronics and Sports) and the negative: *tech support* (-2.59, Electronics), *save your* (-2.6, Video Games), *very thin* (-2.06, Clothes).

The general trigrams also follow a grammatical pattern such as verb plus strong object: *is a great* (2.38 ± 0.32, in all 10 domains) or a general verb phrase related to discontent such as *would have been* (-0.79 ± 0.3, 8 domains), negation bigrams that begin with a subject are also very frequent, ex., *i would not* (-2.27 ± 0.24, in all data sets). Among the domain-specific trigrams we can find *waste of money* (-4.62, Toys), *does not fit* (-2.92, Automotive), *been using this* (1.95, Health), *for the money* (1.49, Electronics).

Another noteworthy result is **the performance of lexicons following grammar patterns**. During our work we have come across the insight that the past tense was an indication of negativity in review while the present or future were positive markers. Interestingly enough the very simple *PF lexicon* consisting of just what the English operators used to introduce a given tense (such as *will, has, must, is* as positive indicators and *was, would, had, were* as negative), scored over the baseline with scores ranging 0.34–0.38. Also other simple lexicons (*i.e.*, Simple List lexicon with positive words such as *good, awesome, great, fantastic, wonderful* and negative ones as *bad, terrible, worst, sucks, awful and dumb*) and their variations scored over the baseline.

5.2.2. Lexicon-based Approach Evaluation

Most of the lexicons achieved a better F-measure than the 33.(3)% random baseline. There were three lexicons that scored more than a 40% F-measure. These were the Bing Liu lexicon and two lexicons generated by us with bigrams and unigrams. This proves that lexicon-based sentiment analysis is strongly domain dependent. The lexicons built particularly for specific domain will achieve higher accuracy than universal lexicons. Consideration should be given to the F-measure of trigrams which is the same as the baseline of 3-class classifications. This can be explained by a very low number of simultaneously high polarized and frequent trigrams in training data.

It is worth pointing out that the Music domain dataset is more difficult, and complex in terms of building sentiment lexicons and achieving satisfactory levels of accuracy. This is the only example when a bigram lexicon is worse than the Bing Liu lexicon, although unigram models are still the best solution.

The average accuracy of publicly available sentiment analysis tools is near 60% [42]. Our generated unigram lexicons achieved an F-measure equal to 50% and it was the simplest version only counting word occurrences in documents. Extending this approach with negation handling, weighting, and even basic rule mining would outperform the average results for existing systems.

5.2.3. Underperforming Lexicons

The Figure 7 presents evaluation of the underperforming lexicons: automatic lexicon generation methods SO-PMI and SO-LSA, and the NRC lexicons [43]. Underperforming means here, that their accuracy is close to random guess. There was no reason to use them in ensemble classification step. We added our frequentiment lexicons for comparison purposes. The additional line in the graphs represents F-measure equal to 0.33%.

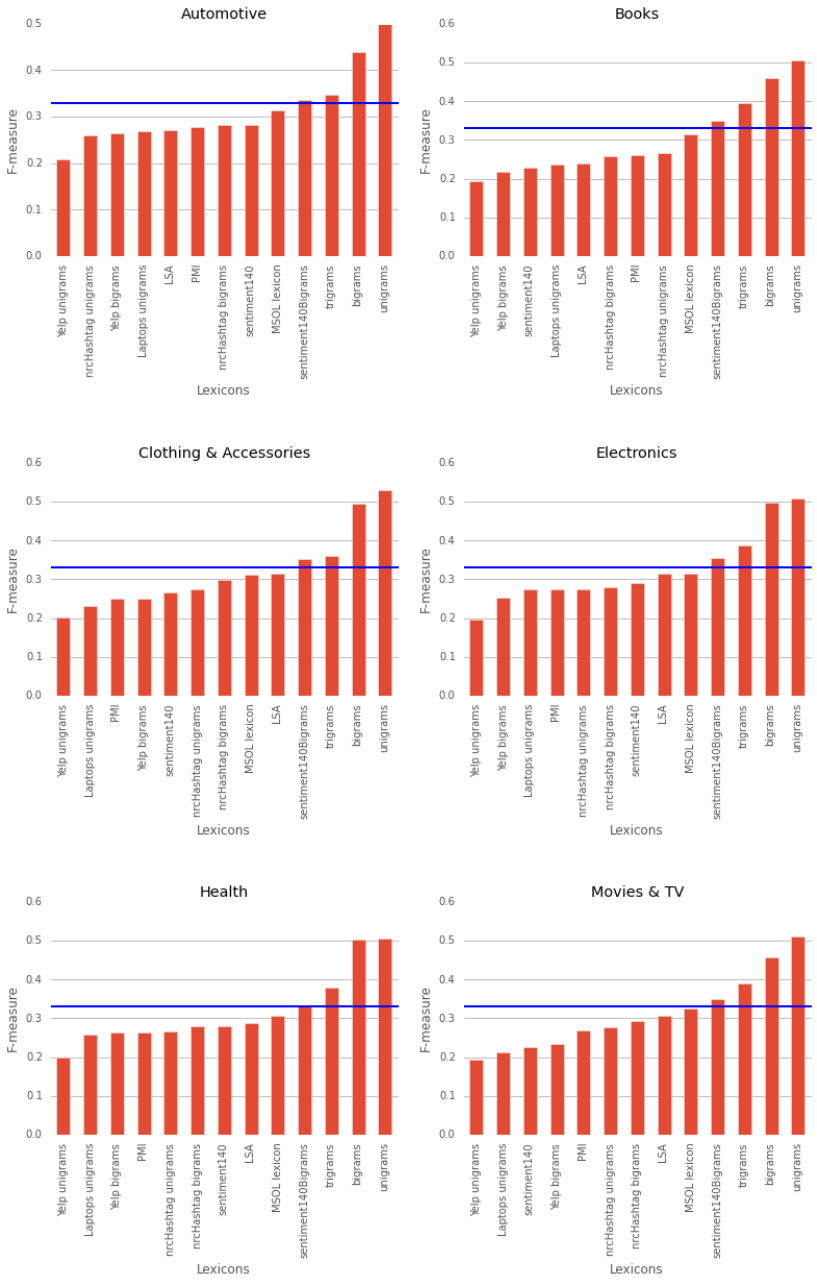


Figure 7. *Cont.*

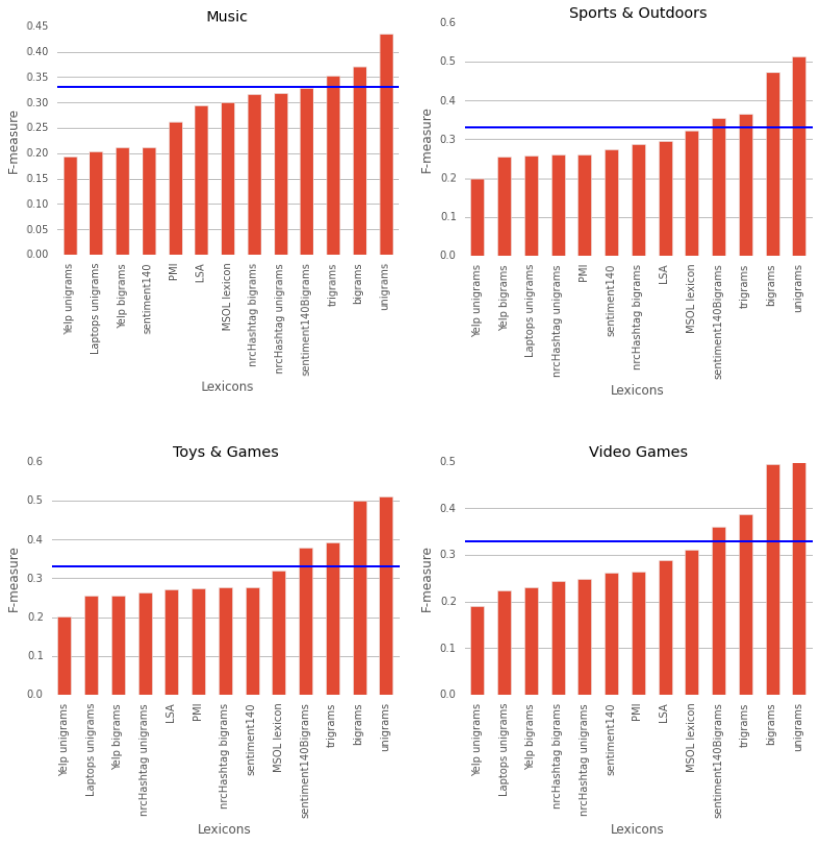


Figure 7. Performance comparison of proposed and underperforming lexicons—F-measure.

We provided this comparison to present the performance of Twitter-based lexicon for longer documents such as reviews and some recognizable lexicons produced for review texts. The NRC team (National Research Council Canada) won SemEval competition in 2013, hence their lexicon is good example for state-of-the-art.

The frequentiment-based lexicons, outperforms all the other tested lexicons. It is worth to mention that these lexicons are smaller, which means faster in terms of required computation time. For example the MSOL Lexicon [44]. contains more than 76,000 ngrams and it gives worse score than the trigram lexicons, that contain on average approximately 100 ngrams. Similarly, the NRC Hashtags with bigrams lexicon consists of more than 300,000 of ngrams.

It is worth pointing out that Amazon laptops lexicon performed poorly, even for Electronics domain. This lexicon was build for specific problem and we think it could be the reason for its inadequate performance in our experiment. Similarly the Yelp Restaurant lexicon didn't perform well neither. It may be understandable, because it was based on restaurant review. These lexicons were used to generate winning submissions for the sentiment analysis shared task of SemEval-2014 Task 4.

Another compared lexicons is Sentiment 140 with unigrams. Unfortunately, it contains words such as Twitter's user names and hashtags. This kind of words do not appear to often in review texts. Hence, the performance of this lexicon is low than it may be expected. The Sentiment 140 lexicons with bigrams achieved quite well accuracy. It is still worse than trigram versions of the frequentiment, however it is always better than random guess. Unfortunately, we decided do not use this lexicon in ensemble classification due to it's size. Sentiment 140 with bigrams consists of more than 670,000 bigrams and even simple counting sentiment based on this lexicon is really time consuming tasks.

Vividly inferior performance of the PMI approach can be attributed to the distribution of bigrams among documents. Words from the initial positive and negative word lists do not occur often with other words. For example features from both these list co-occur with an average number of 302 words, in 1.5 ± 0.5 out of 12k reviews as averaged feature, on the review subset in Automotive reviews.

For example the word best ends up having a very strong negative sentiment orientation (-3.6) in randomly chosen cross-validation fold of Automotive reviews. It is caused by the fact the the only feature from the positive/negative lists it co-occurs with the word "good" in 2 reviews, while the word "best" occurs in 464 reviews and the word "good" occurs in 2192 reviews. On the other hand the fallback minimum co-occurrence with other words is 1.

Similar argument holds when discussing the underperformance of SO-LSA. It is also highly dependent on the selection of the right seed words.

The SO-PMI and SO-LSA histograms of sentiment predictions for each document are presented Figure 8. We see that most of the SO-PMI-based document's predictions across all domains are lower than 0. Hence, even the choosing the best threshold for assigning sentiment orientation for each document doesn't provides valuable measures. I decided add this results for comparison only purposes and do not use it in ensemble classification.

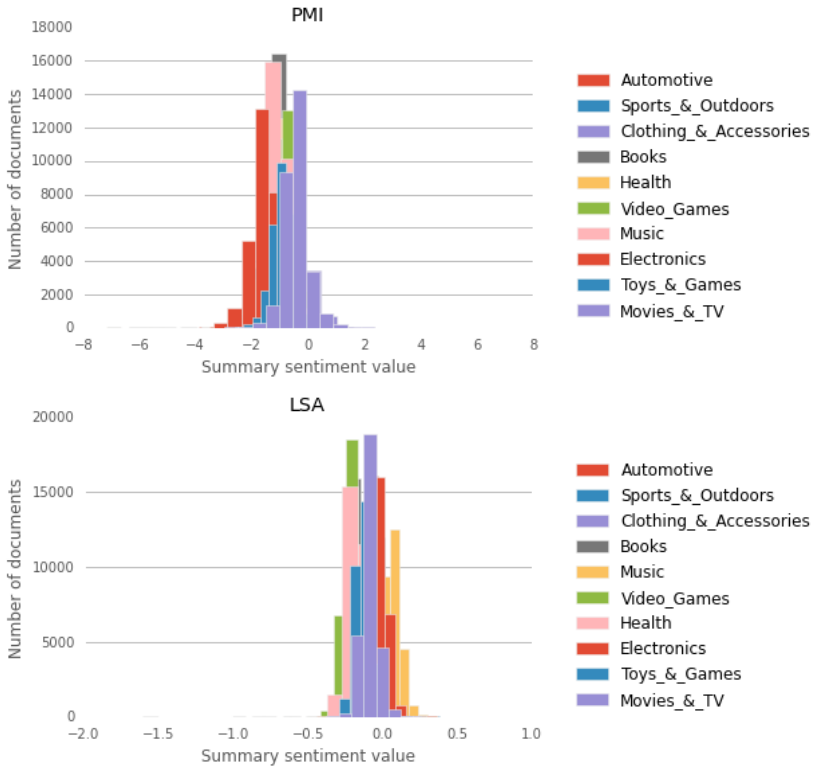


Figure 8. Sentiment predictions for the SO-PMI and SO-LSA—based on first Cross-Validation fold from each domain.

5.3. Comparison Between Domains

In order to omit accumulation of Type I errors in comparisons we used Demsar's scheme for comparing multiple methods across multiple domains. This non-parametric rank-based procedure starts with a null hypothesis that all methods have similar average ranks among problems. We conducted an Iman-Davenport corrected Friedmann test against this hypothesis. The Iman-Davenport procedure calculated the test statistic distributed according to F-distribution with 24 and 216 degrees of freedom.

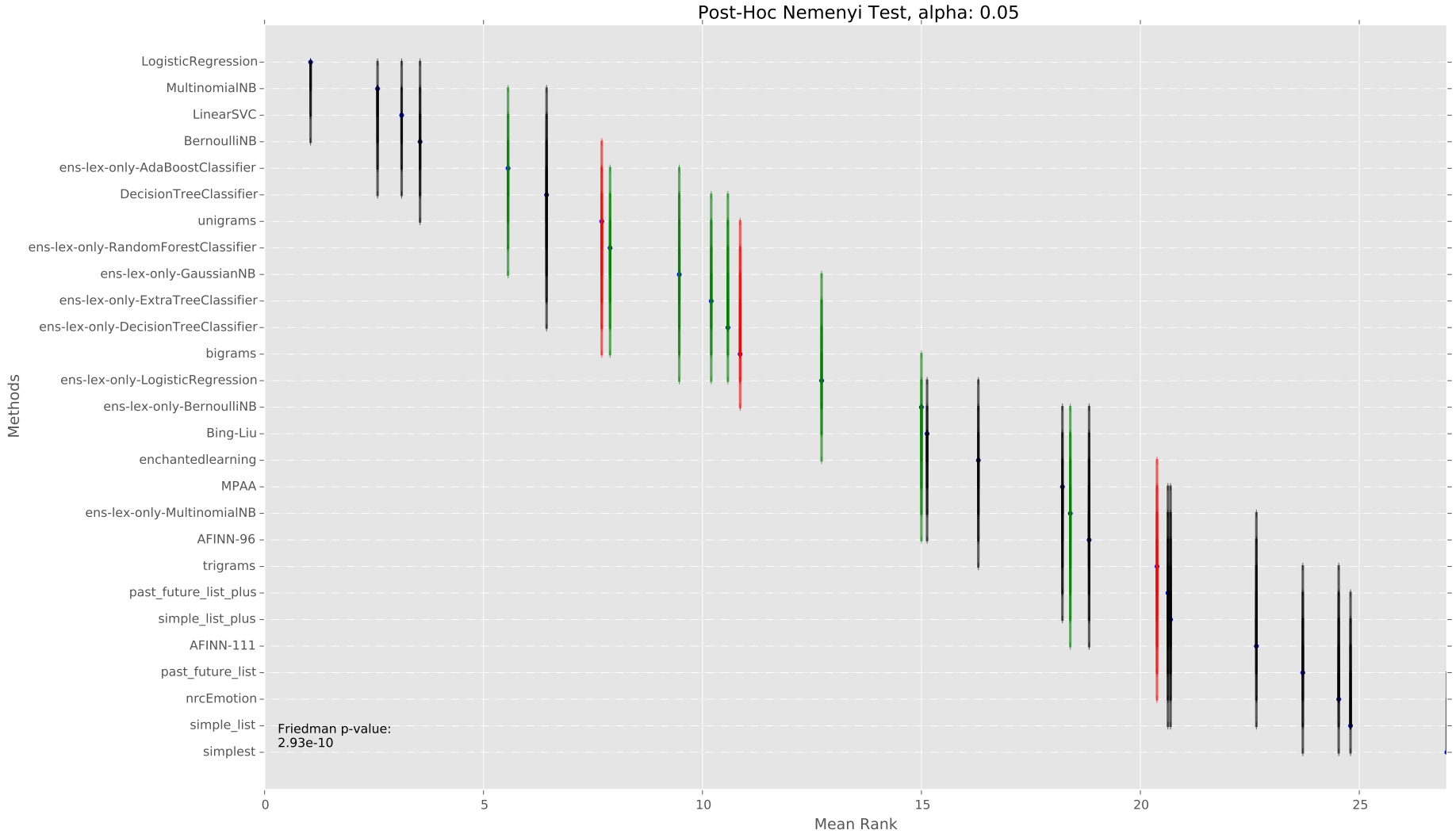


Figure 9. The results of Nemenyi pairwise hypothesis post-hoc procedure.

We discarded the null hypothesis with a p-value $2.93^{-10} < 0.001$. We then conducted the post-hoc Nemenyi test for pairwise comparisons (see Figure 9), which corrects for comparison errors. Its null hypothesis states that two algorithms have the same average ranks. We tested this hypothesis for each pair of algorithms against the Nemenyi test with significance $\alpha = 0.01$.

We noted that the top 4 ranked methods are supervised learners as expected. Our best ensemble method—AdaBoost based on lexicon input—achieved an average rank of 5.5 and is not significantly different than all the supervised approaches apart from the very best Logistic Regression. In addition, the unigram lexicon was ranked 7.7 and was not significantly worse than the higher ranked methods. For example it is not significantly worse than a Bernoulli or a Decision Tree supervised approach, while it remains significantly better than other tested lexicons.

In terms of the best ensemble fusion classifiers, nor is Random Forests (ranked 7.91) placed significantly worse than the majority of the best supervised approaches. The best performing supervised learners are Logistic Regression ranked 1, Multinomial and Bernoulli versions of Naive Bayes (2.6 and 3.6) and Linear SVC 3.1.

Computational Complexity

It is worth pointing out the comparison of memory and time complexity of the examined methods. The simplest and fastest ones are lexicon-based approaches. They need on average less than one minute to compute for all 30,000 reviews. The extension of lexicon-based approaches with an ensemble classifier takes only milliseconds longer than the single lexicon approach. The reason is that a fusion classifier uses only a small feature set (as in the ensemble described in Section 2), hence the training and test phases are time and memory efficient. Computations for lexicons were done in parallel with Python's threading module [45]. The lexicon-based ensemble is more than two times faster than the supervised learning approach (Figure 10). This figure represents an average time execution from 10 folds of cross-validation for each domain. The ensemble lexicon method is a sum for time of frequentiment generation and fusion classifier.

In addition, the SO-PMI and SO-LSA lexicons was added to compare it with frequentiment generation method. The time of execution for frequentiment and SO-PMI is nearly the same; the time of iterating through documents. However, SO-LSA requires more complex computation such as TF/IDF matrices *etc.*, hence the overall time complexity is much higher. It can be seen in Figure 10 with logarithmic scale for y axis.

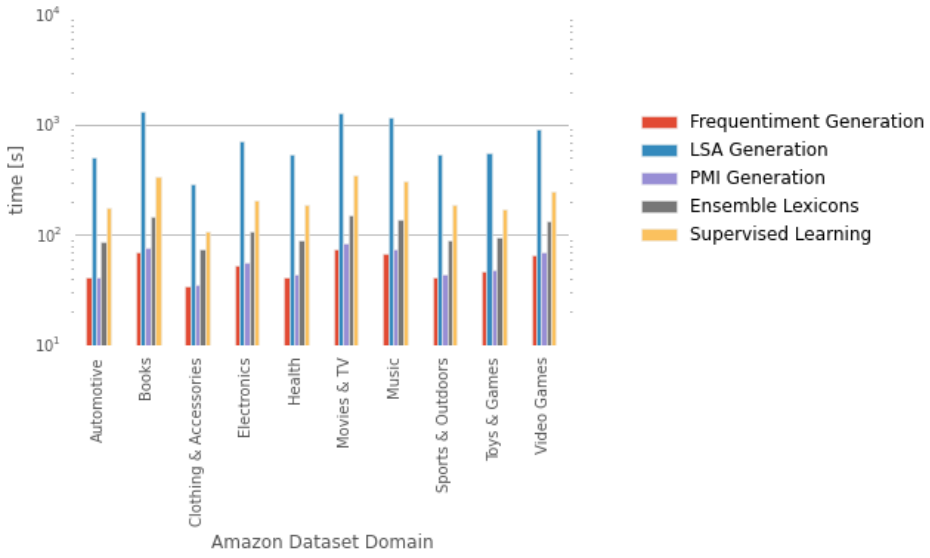
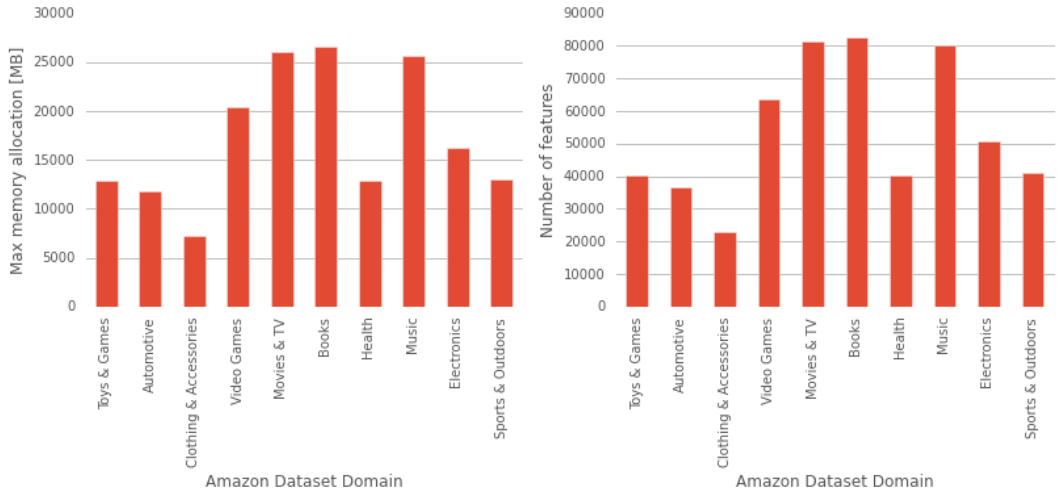


Figure 10. Average time execution of examined methods.

In addition, the memory complexity of the ensemble method is much more efficient than the supervised method. It can be seen in Figure 11a where the GBs of RAM memory is the result of storing a big matrix with all ngrams in memory. Lexicons used only a couple of MBs of memory because the computation was done for each document separated and only the lexicon's predictions for the fusion classifier were stored in memory. The memory allocation was investigated by the Python memory profiler [46]. This is a module for monitoring memory consumption of a process as well as line-by-line analysis of memory consumption for python programs. It doesn't analyze the maximum used memory (memory peaks), but only the difference between memory allocation before and after execution of each line of code. Hence, we didn't see the maximum RAM memory usage, which was for some domains more than 120GB of RAM.



(a) **(b)**

Figure 11. Memory complexity and feature size of examined supervised approach. (**a**) Average memory allocation during the experiment on the supervised learning approach; (**b**) Average feature size of supervised learning methods.

6. Discussion

Lexicon-based methods in general are time-efficient and inaccurate in cases of sophisticated opinion texts. Also, lexicons may not scale well for specialized texts and are domain specific by nature. Medhat *et al.* [7] noted that the dictionary based approach may fail to find opinion words with domain and context specific orientations. The corpus generation approach addresses this issue yet used alone may not be as effective as the dictionary-based approach. On the other hand supervised learning methods are in general more accurate, but much slower than lexicon-based methods. Real world applications usually prefer an approach that provide a trade-off between these two conflicting optimization targets. Ensemble learning is a compromise approach between effectiveness and accuracy and that's the reason why we conducted such an analysis.

Our extensive comparison confirms some of these remarks while counters others (see Table 6). The proposed unigram frequentiment approach managed to achieve significantly better performance than established dictionary-based ones such as AFINN, MPAA, Bing Liu's or NRC Canada's lexicons, by leveraging general and domain-specific features in the generated lexicon.

Table 6. Best thresholds for frequentiment F-measure.

	Unigrams	**Bigrams**	**Trigrams**
Automotive	1.4	1.3	0.1
Books	1.4	1.3	0.4
Clothing & Accessories	1.5	1.7	0.1
Electronics	1.7	1.8	0.1
Health	1.2	1.6	0.1
Movies & TV	1.4	1.2	0.3
Music	1.4	1.1	0.2
Sports & Outdoors	1.5	1.5	0.1
Toys & Games	1.5	2	0.2
Video Games	1.3	1.9	0.3
Avg $\pm$ Std	1.43 ± 0.13	1.54 ± 0.29	0.19 ± 0.10

It has also managed to overcome weaknesses in the dictionary approaches where some features may be badly classified. For example the word *refund* is a strong marker of negative sentiment in all evaluated review domains, while Bing Liu's dictionary lists it as a positive word. Also frequentiment lexicons are smaller in size and better at selecting proper words, which is not only a good property in itself for any lexicon, but also a problem in the case of dictionary approaches. This problem can be observed, for example, in the AFINN data sets, where the older and smaller AFFIN-96 outperformed, although not with statistical significance, its newer and extended version—AFFIN-111. The MPAA lexicon proves the quite good accuracy across all domains.

We also noted that generated bigram approaches, which took negations into account, performed better than the dictionary approaches, while not significantly better than Bing Liu, MPAA and EnchantedLearning lexicon. They were significantly better than all other lexicons including both versions of the AFFIN.

The last of the generated lexicons—the trigram frequentiment lexicons were significantly similar to AFFIN lexicons while being up to 50 times smaller in size.

We confirmed statistically the preliminary results of our conference paper [27], which considered fewer data sets and where only the top 5 and top 25 sentiment markers from each frequentiment lexicon were been selected.

We also confirmed Medhat *et al.*'s remarks about supervised learners, which were the best among evaluated methods, yet two of the proposed ensemble approaches achieved significantly comparable performance levels while remaining up to 3–5 times faster than the supervised approaches.

We also address Whitehead and Yaeger's [14] problems of verifying the significance of ensemble approaches. We show that AdaBoosting and Random Forests perform significantly better than the others, and on a par with supervised learners. Whitehead and Yaeger pointed out that there are problems with AdaBoost overfitting, but we used the same parameters and avoided this problem. We believe this is due to the fact that we used a differentiated set of inputs from lexicons and using frequentiment instead of an odds ratio to build generated lexicons. They also noted small improvements of 3 percentage points in ensembles over their generated odds ratio-based lexicons. The improvement our best ensemble methods provides over the best generated frequentiment lexicon is of the magnitude of less than 1 percentage point in case of the easiest problems (*i.e.*, the ones where all methods performed well, like Clothes and Accessories) up to 7.5 percentage points in the hardest problems (ex. Music).

7. Conclusions and Future Work

We propose a new method for lexicon generation—**frequentiment**—based on likelihood increased, when the document contains a given feature averaged by score per feature. We provide a scheme for sentiment annotation based on frequentiment lexicons and describe optimal parameter

selection for the process. We have shown interesting insights and example markers generated by frequentiment lexicons. The unigram frequentiment lexicons yielded best scores among lexicons while being consistently smaller than well-established dictionary-based lexicons. The bigram frequentiment lexicons performed on a par with the best dictionary-based lexicons—Bing Liu, MPAA and Enchanted Learning, while remaining smaller. The trigram lexicons are comparable to AFFIN dictionary lexicons in performance while remaining up to 50 times smaller.

We also proposed an ensemble approach based on lexicon input, where lexicons served as weak classifiers and different fusion classifiers were used. We conclude that AdaBoosting performed the best among all fusion classifiers and was not significantly worse than the best, baseline supervised methods. Random Forests learner was also a well performing fusion classifier. While AdaBoost ranked in between supervised approaches, Random Forest ranked right after them.

We noted the remarkable performance of the unigram frequentiment lexicons, especially given their size and the fact that in unigram classification negations were not taken into account. The comprehensive comparison of several well established lexicons was presented. The Twitter based lexicons proved to be questionable choice for longer document's such as reviews.

The next steps related to the experiments presented are: repeat the analysis over bigger number of domains and new data sets and run it for heavily inflected languages, such as Polish. In addition, experiments with other methods of ensemble classification would be a great extension. Extension of the lexicon with emoticons and emoji will be also investigated. Afterwards, we will compare our methods with emoticons and emojis to some well-known lexicons such as [17,47]. In addition, other seed words for SO-LSA method may be investigated. Moreover, evaluation of Twitter data will be needed, because a lot of experiments for lexicon generation have been performed in this area. It will be more consistent and comprehensive to compare methods used on the same data.

Acknowledgments: This paper is a continuation of [27]. The authors would like to acknowledge the help of Joanna Kaczmar in constructing the frequentiment measure and implementing Python code. The work was partially supported by Fellowship co-financed by European Union within European Social Fund; The European Commission under the 7th Framework Programme, Coordination and Support Action, Grant Agreement Number 316097 [ENGINE]; The National Science Centre the research project 2014-2017 decision no. DEC-2013/09/B/ST6/02317.

Author Contributions: Wrocław, Poland, 18 December 2015. Łukasz Augustyniak, Piotr Szymański, Włodzimierz Tuligłowicz and Tomasz Kajdanowicz designed the experiments; Łukasz Augustyniak, Piotr Szymański performed the experiments; Łukasz Augustyniak, Piotr Szymański and Tomasz Kajdanowicz analyzed the data; Łukasz Augustyniak, Piotr Szymański and Włodzimierz Tuligłowicz wrote the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tumasjan, A.; Sprenger, T.O.; Sandner, P.G.; Welpe, I.M. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *ICWSM* **2010**, *10*, 178–185.
2. Ghiassi, M.; Skinner, J.; Zimbra, D. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Syst. Appl.* **2013**, *40*, 6266–6282.
3. Brody, S.; Elhadad, N. An Unsupervised Aspect-sentiment Model for Online Reviews. In Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, CA, USA, 1–6 June 2010; pp. 804–812.
4. Bollen, J.; Mao, H.; Zeng, X. Twitter mood predicts the stock market. *J. Comput. Sci.* **2011**, *2*, 1–8.
5. Liu, B.; Zhang, L. A Survey of Opinion Mining and Sentiment Analysis. In *Mining Text Data*; Springer-Verlag: New York, NY, USA, 2012; pp. 415–463.
6. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135.
7. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113.
8. Liu, B. Sentiment Analysis and Opinion Mining. In Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AAAI-11), San Francisco, CA, USA, 7–11 August 2011.

9. Hu, M.; Liu, B. Mining and Summarizing Customer Reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; ACM: New York, NY, USA, 2004; pp. 168–177.
10. Qiu, G.; Liu, B.; Bu, J.; Chen, C. Expanding Domain Sentiment Lexicon Through Double Propagation. In Proceedings of the 21st International Jont Conference on Artifical Intelligence (IJCAI'09), Pasadena, CA, USA, 11–17 July 2009; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2009; pp. 1199–1204.
11. Hatzivassiloglou, V.; McKeown, K.R. Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, 7–12 July 1997; pp. 174–181.
12. Yu, H.; Hatzivassiloglou, V. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP'03), Sapporo, Japan, 11–12 July 2003; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; pp. 129–136.
13. Read, J.; Carroll, J. Weakly supervised techniques for domain-independent sentiment classification. In Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, Hong Kong, China, 6 November 2009; ACM: New York, NY, USA, 2009; pp. 45–52.
14. Whitehead, M.; Yaeger, L. Sentiment mining using ensemble classification models. In *Innovations and Advances in Computer Sciences and Engineering*; Springer Netherlands: Dordrecht, The Netherlands, 2010; pp. 509–514.
15. Lu, Y.; Castellanos, M.; Dayal, U.; Zhai, C. Automatic Construction of a Context-aware Sentiment Lexicon: An Optimization Approach. In Proceedings of the 20th International Conference on World Wide Web (WWW'11), Hyderabad, India, 30 March 2011; ACM: New York, NY, USA, 2011; pp. 347–356.
16. Ding, X.; Liu, B.; Yu, P.S. A Holistic Lexicon-based Approach to Opinion Mining. In Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM'08), Palo Alto, CA, USA, 11–12 February 2008; ACM: New York, NY, USA, 2008; pp. 231–240.
17. Mohammad, S.M.; Kiritchenko, S.; Zhu, X. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, GA, USA, 14–15 June 2013; Association for Computational Linguistics: Stroudsburg, PA, USA, 2013; pp. 321–327.
18. Weiss, S.M.; Kulikowski, C.A. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1991.
19. Bo, P.; Lillian, L.; Shivakumar, V. Thumbs up?: Sentiment classification using machine learning techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, USA, 6–7 July 2002; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; Volume 10, pp. 79–86.
20. Ye, Q.; Zhang, Z.; Law, R. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Syst. Appl.* **2009**, *36*, 6527–6535.
21. Schler, J. The Importance of Neutral Examples for Learning Sentiment. In Proceedings of the Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations (FINEXIN05), Ottawa, ON, Canada, 26–27 May 2005.
22. Bai, X. Predicting consumer sentiments from online text. *Decis. Support Syst.* **2011**, *50*, 732–742.
23. Socher, R.; Perelygin, A.; Wu, J.Y.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), Seattle, WA, USA, 18–21 October 2013; Volume 1631, p. 1642.
24. Narayanan, V.; Arora, I.; Bhatia, A. Fast and Accurate Sentiment Classification Using an Enhanced Naive Bayes Model. In Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2013), Hefei, China, 20–23 October 2013; Volume 8206, pp. 194–201.

25. Gamon, M. Sentiment Classification on Customer Feedback Data: Noisy Data, Large Feature Vectors, and the Role of Linguistic Analysis. In Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), Geneva, Switzerland, 23–27 August 2004; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004.
26. Augustyniak, L.; Kajdanowicz, T.; Kazienko, P.; Kulisiewicz, M.; Tuliglowicz, W. An Approach to Sentiment Analysis of Movie Reviews: Lexicon Based *vs.* Classification. In Proceedings of the 9th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2014), Salamanca, Spain, 11–13 June 2014; pp. 168–178.
27. Augustyniak, L.; Kajdanowicz, T.; Szymanski, P.; Tuliglowicz, W.; Kazienko, P.; Alhajj, R.; Szymanski, B.K. Simpler is better? Lexicon-based ensemble sentiment classification beats supervised methods. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), Beijing, China, 17–20 August 2014; pp. 924–929.
28. Whitehead, M.; Yaeger, L. Sentiment Mining Using Ensemble Classification Models. In *Innovations and Advances in Computer Sciences and Engineering*; Springer Netherlands: Dordrecht, The Netherlands, 2010; pp. 509–514.
29. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140.
30. Schapire, R.E. A Brief Introduction to Boosting. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99), Stockholm, Sweden, 31 July–6 August 1999; pp. 1401–1406.
31. Polikar, R. Ensemble Based Systems in Decision Making. *IEEE Circ. Syst. Mag.* **2006**, *6*, 21–45.
32. McAuley, J.; Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; ACM: New York, NY, USA, 2013; pp. 165–172.
33. Python Library Beautiful Soup. Available online: https://pypi.python.org/pypi/beautifulsoup4 (accessed on 1 February 2015).
34. Python Library Unidecode. Available online: https://pypi.python.org/pypi/Unidecode (accessed on 1 February 2015).
35. Turney, P.D.; Littman, M.L. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Trans. Inf. Syst.* **2003**, *21*, 315–346.
36. Nielsen, F.Å. AFINN Informatics and Mathematical Modelling, Technical University of Denmark, 2011. Available online: http://www2.imm.dtu.dk/pubdb/p.php?6010 (accessed on 24 December 2015).
37. Wilson, T.; Wiebe, J.; Hoffmann, P. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 347–354.
38. Mohammad, S.M.; Turney, P.D. Crowdsourcing a Word-Emotion Association Lexicon. *Comput. Intell.* **2013**, *29*, 436–465.
39. Kiritchenko, S.; Zhu, X.; Cherry, C.; Mohammad, S. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; Association for Computational Linguistics and Dublin City University: Dublin, Ireland, 2014; pp. 437–442.
40. Turney, P.D.; Littman, M.L. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.* **2003**, *21*, 315–346.
41. SemEval Contest Website. Available online: https://www.cs.york.ac.uk/semeval-2013/task2 (accessed on 1 March 2015).
42. Cieliebak, M.; Dürr, O.; Uzdilli, F. Meta-Classifiers Easily Improve Commercial Sentiment Detection Tools. In Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; European Language Resources Association (ELRA): Reykjavik, Iceland, 2014.
43. NRC Canada Website. Available online: http://saifmohammad.com/WebPages/lexicons.html (accessed on 1 August 2015).
44. Macquarie Semantic Orientation Lexicon (MSOL) Link. Available online: http://www.saifmohammad.com/Release/MSOL-June15-09.txt (accessed on 1 August 2015).

45. Higher-level threading interface in Python. Available online: https://docs.python.org/2/library/threading.html (accessed on 1 February 2015).
46. Python Library Memory Profiler. Available online: https://pypi.python.org/pypi/memory/profiler (accessed on 1 August 2015).
47. Tang, D.; Wei, F.; Qin, B.; Zhou, M.; Liu, T. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. In Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014), Dublin, Ireland, 23–29 August 2014; Dublin City University and Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 172–182.