LABR: A Large Scale Arabic Book Reviews Dataset

Mahmoud Nabil, Mohammed Aly, Amir Atyia

***Abstract*—Opinions on the Internet have a great influence on our own decisions when we plan to buy a product, travel abroad or even read a book. This is because these activities consume our worthy resources in terms of time and money. In this paper we introduce LABR, the largest sentiment analysis dataset to-date for the Arabic language. We extend the work done in ? by developing a comprehensive simulations and analysis on LABR (Large Scale Arabic Book Reviews Dataset). In particular, we perform here an extended survey of the different classifiers typically used for the sentiment polarity classification problem.**

I. Introduction

Opinion mining is gaining a large attention nowadays. By means of social networks any one can share opinions or ideas with his peers in no time, making activities like shopping on-line, reading a book, watching a movie, or estimating the popularity of a public figure be influenced other people's sentiments towards these entities. Also the last decade witnessed an explosion in the number of social media platforms and the number of people using them.

Opinion mining is the science of extracting emotions and opinions from raw text reviews. It can be categorized to sentiment classification and feature-based opinion mining [**?**]. The goal of sentiment classification is to analyze the sentiment (positive, negative, and neutral) towards the main entity of the sentence. In feature-based opinion mining the goal is to identify the main entity in the review or analyze the attitude towards a certain aspect of the review.

A lot of work has been proposed that target most of the challenging aspects of the sentiment analysis task, [**?**] discuss some of these challenges. These challenges are the same for most languages with some specific challenges to other languages.

Most of the work done in sentiment analysis and the data sets gathered targets English language with very little work on Arabic. One of the reasons is the prevalence of the English websites where 55% of the visited websites on the Internet use English while only 0.8% uses Arabic[1]. Another reason is the complexities of the Arabic language and different Arabic dialects exists in each Arab counry beside different Arabic dialects exist in the same country. In this work, we try to address the lack of large-scale Arabic sentiment analysis datasets in this field, in the hope of sparking more interest in research in Arabic sentiment analysis and related tasks. Towards this end, we introduce **LABR**, the **L**arge-scale **A**rabic **B**ook **R**eview dataset. It is a set of over 64K book reviews, each with a rating of 1 to 5 stars.

The contributions in this paper can be summarized as:

1) We present the largest Arabic sentiment analysis dataset to-date (up to our knowledge).
2) We provide standard splits for the dataset into training, validation and testing sets. This will make comparing different results much easier. We will make the dataset and the splits publicly available.
3) Application of a wide range of classifiers to athe large set of book reviews that we collected.

II. Market and Related Work

A. Industry and Market

To have an idea of competing sentiment analysis products, we review here the existing market. Egyptian products include 25trends[2] , which analyzes posts from Twitter and Facebook and performs sentiment analysis. Unfortunately, the demo service does not perform satisfactorily. Also Repustate's sentiment analysis API target Arabic language, but its performance is weak[3]. Table 1 shows examples of some available products and their features/capabilities.

B. Related Work

According to [**?**] sentiment analysis is handled by either lexicon-based approaches, machine learning approaches like text classification tasks, or hybrid approaches.

For lexicon-based approaches [**?**] developed a Semantic Orientation CALculator and used some annotated dictionaries of words where the annotation covers the word polarity and strength. They used Amazon's Mechanical Turk service to collect validation data to their dictionaries and based their experiments on four different corpora with equal numbers of positive and negative reviews. [**?**] and [**?**] used a sentiment

[1]Languages used on the Internet

[2]www.25trends.me

[3]www.repustate.com/api-demo/

Company	Language		Features
	Arabic	English	
HP's Autonomy		✓	
IBM's Smarter Analytics		✓	Twitter, Facebook
Sentiment140		✓	Twitter
twitrratr		✓	Twitter
Social Mention		✓	Twitter ...
tweetfeel		✓	Twitter
Repustate's	✓	✓	Twitter, Facebook, ...
25trends	✓	✓	Facebook, Twitter, YouTube

Table I: Table 1: Products and their features.

lexicon that depends on the context of every polarity word (contextualized sentiment lexicon) and based there experiments on customer reviews form Amazon and TripAdvisor[4].

In general lexicon-based sentiment classifiers show a positive bias [**?**], however [**?**] implemented normalization techniques to overcome this bias.

For machine learning approaches [**?**] used part of speech and n-grams to build a sentiment classifiers using the Multinomial Naive Bayes classifier, SVM and conditional random fields. They tested their classifiers on a set of hand annotated twitter posts. [**?**] proposed an approach to target dependent features in the review by incorporating synaptic features that are related to the sentiment target of the review. They build binary SVM classifier to perform the classification of two tasks: subjectivity classification and polarity classification.

For hybrid approaches, [**?**] used n-gram features, lexicon features, and part of speech to build an Ada-boost classifier. They used three different corpora of Twitter messages (HASH, EMOT and iSieve) to evaluate their system. [**?**]constructed a domain specific lexicon and used it to back the classification of the reviews. They used a data set for customer reviews from TripAdvisor. For Arabic little work has been proposed the sentiment analysis problem. [**?**] perform a multilingual sentiment analysis of English and Arabic Web forums. [**?**] proposed the SAMAR system that perform subjectivity and sentiment analysis for Arabic social media using some Arabic morphological features. Some Arabic sentiment data sets have been collected as mentioned in Tabel II.

III. Sentiment Analysis Challenges

Sentiment analysis is still a formidable natural language processing task **?** because unlike text categorization where the tokens depends largely on the domain or the category, in sentiment analysis we usually have three semantic orientations (positive, negative, and neutral) and most tokens can exist in the three categories at the same time. Another reason is the language ambiguity where one or more polarity token depends on the context of the sentence. Also most Internet users tend to give a positive rating even if their reviews contain some misgivings about the entity, or some sort of sarcastic remarks, where the intent of the user is the opposite of the written text.

Some challenges are specific to Arabic language such as few research **?**; **?**; **?**; **?**; **?**, and very few datasets available for natural language different processing tasks. In addition, the complexities of the Arabic language, due to Arabic being a morphologically rich language, add a level of complication. Another problem is the existence of Modern Standard Arabic side by side with different Arabic dialects, which are not yet standardized. **?** presented some other challenges for Arabic language such as: the unavailability colloquial Arabic parsers this problem faces all the solutions that depened on the parsing strucrure of the sententce, another challenge is the need for person name recognition as some Arabic names are derived from adjectives, also compound phrases that are widely used in Arabic where the sentiment of the whole compound phrase cannot be determined by it's constituent words.

IV. Dataset Collection

We downloaded over 220,000 reviews from the book readers social network www.goodreads.com during the month of March 2013. These reviews were from the first 2143 books in the list of *Best Arabic Books*. After harvesting the reviews, we found out that over 70% of them were not in Arabic, either because some non-Arabic books exist in the list, or because of existing translations of some of the books in other languages. After filtering out the non-Arabic reviews, and performing several pre-processing steps, we ended up with 64,245 Arabic reviews. Due to a limitation in the API provided by the host website, we were able to retrieve only the first 300 characters of each review. Upon inspection of the downloaded ones, we found out that only about 20K of them were over 300 characters.

V. Dataset Properties

The dataset contains 64,245 reviews that were submitted by 16,577 users for 2,143 different books. Table III contains some important facts about the dataset and Fig. V.1 shows the number of reviews for each rating. The number of positive reviews is much larger than that of negative reviews. We believe this is because the books we got reviews for were the

[4]Trip Advisor

Data Set Name	Size	Source	Type	Cite
TAGREED (TGRD)	3015	Tweets	MSA/Dialectal	?
TAHRIR (THR)	3008	Wikipedia TalkPages	MSA	?
MONTADA (MONT)	3097	Forums	MSA/Dialectal	?
OCA(Opinion Corpus for Arabi)	500	Movie reviews	Dialectal	?
AWATIF	2855	Wikipedia TalkPages/Forums	MSA/Dialectal	?
LABR(Large Scale Arabic Book Reviews)	63,257	GoodReads reviews[5]	MSA/Dialectal	?

Table II: Arabic sentiment data sets

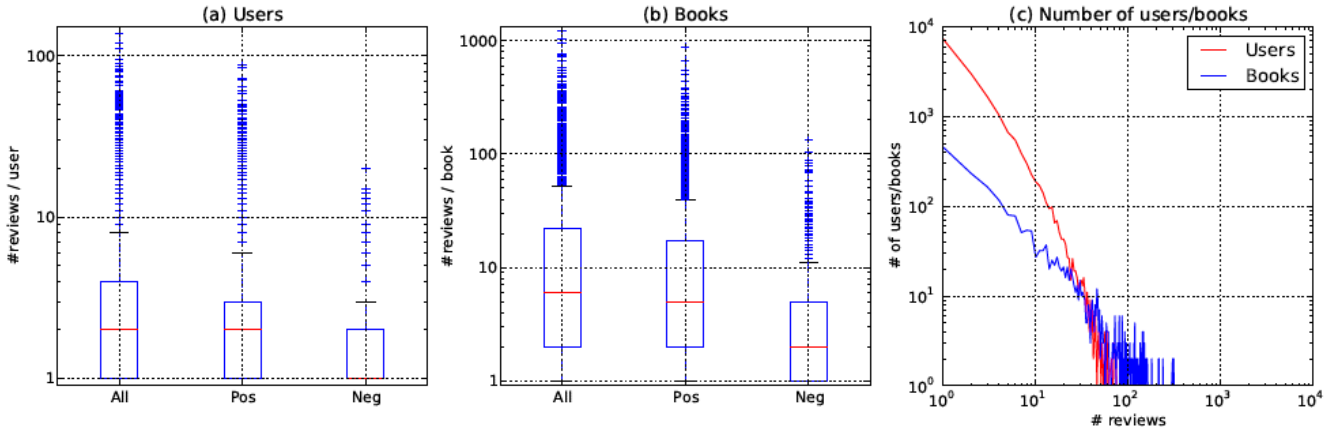

Figure V.2: **Users and Books Statistics**. *(a)* Box plot of the number of reviews per user for all, positive, and negative reviews. The *red* line denotes the median, and the edges of the box the *quartiles*. *(b)* the number of reviews per book for all, positive, and negative reviews. *(c)* the number of books/users with a given number of reviews.

Number of reviews	63,257
Number of users	16486
Avg. reviews per user	3.84
Median reviews per user	2
Number of books	2,131
Avg. reviews per book	29.68
Median reviews per book	6
Median tokens per review	33
Max tokens per review	3,736
Avg. tokens per review	65
Number of tokens	4,134,853
Number of sentences	342,199

Table III: Important Dataset Statistics.

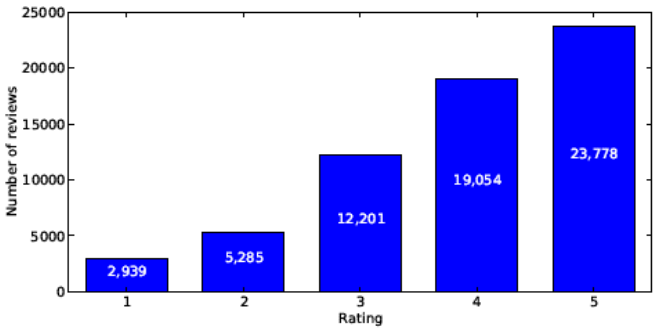

Figure V.1: **Reviews Histogram**. The plot shows the number of reviews for each rating.

most popular books, and the top rated ones had many more reviews than the the least popular books.

The average user provided 3.88 reviews with the median being 2. The average book got 30 reviews with the median being 6. Fig. V.2 shows the number of reviews per user and book. As shown in the Fig. V.2*c*, most books and users have few reviews, and vice versa. Figures V.2*a-b* show a box plot of the number of reviews per user and book. We notice that books (and users) tend to have (give) positive reviews than negative reviews, where the median number of positive reviews per book is 5 while that for negative reviews is only 2 (and similarly for reviews per user). Figure VI.1 shows some examples from the data set.

Fig. V.3 shows the statistics of tokens and sentences. The reviews were tokenized and rough sentence counts were computed. The average number of tokens per review is 34.7, the average number of sentences per review is 3.5, and the average number of tokens per sentence is 9. Figures V.3*a-b* show that the distribution is similar for positive and negative reviews. Fig. V.3*c* shows a plot of the frequency of the tokens in the vocabulary in a log-log scale, which conforms to Zipf's law **?**.

VI. Experiments

In this paper we take the preliminary work **?** as a starting point, and develop comprehensive simulations and analysis. In particular, we perform here an extended survey of the different classifiers typically used for the sentiment polarity

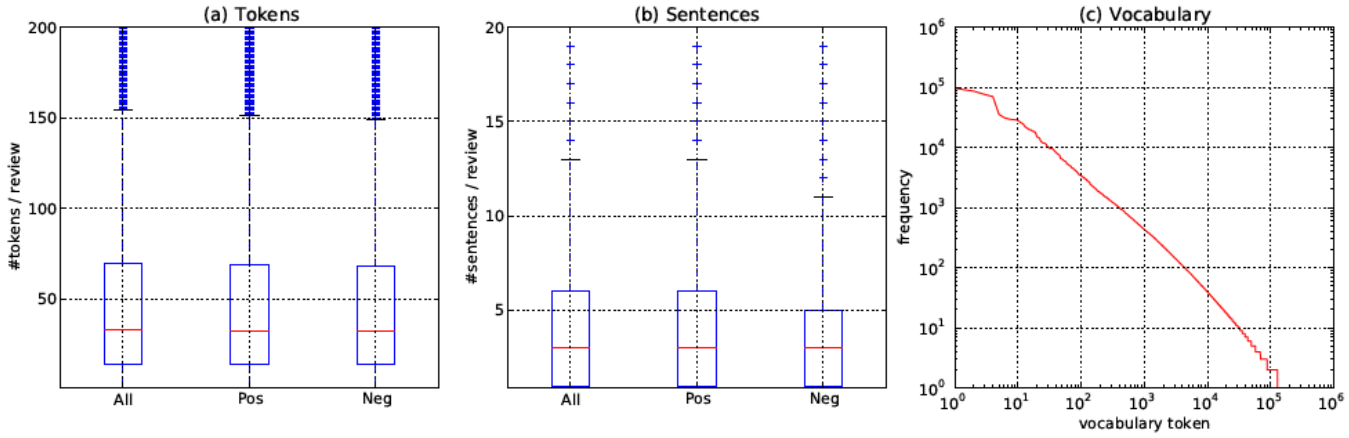

Figure V.3: Tokens and Sentences Statistics . *(a)* the number of tokens per review for all, positive, and negative reviews. *(b)* the number of sentences per review. *(c)* the frequency distribution of the vocabulary tokens.

		Balanced			UnBalanced		
		Positive	Negative	Neutral	Postive	Negative	Neutral
Reviews Count	Train Set	4,936	4,936	4,936	34,231	6,534	9,841
	Test Set	1,644	1,644	1,644	8,601	1,690	2,360
	Validation Set	1,644	1,644	1,644	8,511	1,683	2,457
Features Count	unigrams	115,713			209,870		
	unigrams+bigrams	729,014			1,599,273		
	unigrams+bigrams+trigrams	1,589,422			3,730,195		

Table IV: Data preparation statistics summary

classification problem. In addition we explore the effect of using feature selection, as well as more sophisticated classifiers not generally used in NLP tasks. All Experiments are applied to the data set after applying a standard partitioning to it.

Figure VI.1: LABR reviews examples

A. Data Preparation

In order to test the proposed approaches thoroughly, we partition the data into training, validation and test sets. The validation set is used as a mini-test for evaluating and comparing models for possible inclusion into the final model. The ratio of the data among these three sets is 6:2:2 respectively.

We extend the work in [**?**] by adding a class for neutral reviews. In particular, the data is divided into three classes (positive, negative, and neutral) where ratings of 4 and 5 are mapped to positive, rating of 3 is mapped to neutral, and ratings 1 and 2 are mapped to negative. We constructed two sets of data. The first one is the balanced data set, where the number of reviews are equal in each class category, by setting the size of the class to the minimum size of the three classes. The second one is the unbalanced data set, where the number of reviews are not equal, and their proportions are as exist in the collected data set. Figure VI.2 shows the number of reviews in every class for both balanced and unbalanced sets, and Figure VI.3 and Table IV show the statistic of the number of features for uni-grams range, bi-grams and trigrams range.

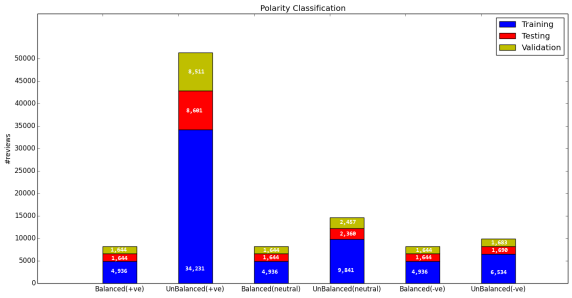

Figure VI.2: Number of reviews for each class category

B. Sentiment Analysis

In this experiment a wide range of standard classifiers are applied to both the balanced and unbalanced datasets using n-gram range of unigrams, bigrams and trigrams where the n-gram range of N degree is a combination of all lower n-grams starting from unigrams, bigrams till the degree N. For example the trigram range is a combination of unigrams, bigrams and trigrams. Also the experiment is applied on the Tf-Idf VI.1 (token frequency inverse document frequency) of the n-grams. Table V shows the result for each classifier on

Features	Tf-Idf	Balanced			Unbalanced		
		1g	1g+2g	1g+2g+3g	1g	1g+2g	1g+2g+3g
MNB	No	0.558/0.560	0.573/0.577	0.572/0.577	0.706/0.631	0.705/0.609	**0.706**/0.612
	Yes	0.567/0.570	0.581/0.584	0.582/0.586	0.680/0.551	0.680/0.550	0.680/0.550
BNB	No	0.515/0.495	0.507/0.473	0.481/0.429	0.659/0.573	0.674/0.553	0.678/0.550
	Yes	0.356/0.236	0.341/0.189	0.338/0.181	0.680/0.550	0.680/0.550	0.680/0.550
SVM	No	0.535/0.534	0.568/0.565	0.570/0.566	0.698/0.690	**0.727/0.712**	**0.731/0.712**
	Yes	0.566/0.564	0.590/0.588	0.589/0.588	**0.734/0.709**	**0.750/0.723**	**0.751/0.725**
Passive Aggressive	No	0.402/0.348	0.489/0.486	0.521/0.525	0.638/0.653	0.693/0.692	0.692/0.676
	Yes	0.504/0.508	0.571/0.574	0.584/0.582	0.681/0.676	**0.740/0.722**	**0.740/0.715**
SGD	No	0.458/0.454	0.459/0.454	0.459/0.455	0.687/0.578	0.687/0.579	0.680/0.570
	Yes	0.416/0.390	0.380/0.292	0.360/0.236	0.680/0.550	0.680/0.550	0.673/0.541
Logistic Regression	No	0.570/0.568	0.586/0.583	0.590/0.585	**0.728/0.707**	**0.743/0.717**	**0.737/0.703**
	Yes	0.587/0.583	0.590/0.588	0.586/0.585	**0.727/0.672**	**0.720**/0.659	**0.709**/0.640
Perceptron	No	0.389/0.328	0.424/0.375	0.449/0.418	0.683/0.680	**0.720/0.705**	**0.719/0.693**
	Yes	0.500/0.502	0.536/0.538	0.526/0.523	0.675/0.672	**0.732/0.714**	**0.726/0.708**
KNN	No	0.428/0.416	0.412/0.395	0.398/0.382	0.675/0.582	0.676/0.577	0.673/0.567
	Yes	0.471/0.461	0.497/0.484	0.490/0.477	0.698/0.619	**0.701**/0.625	0.697/0.615

Table V: **Experiment 1: Polarity Classification Experimental Results**.*Tf-Idf* indicates whether tf-idf weighting was used or not. *MNB* is Multinomial Naive Bayes, *BNB* is Bernoulli Naive Bayes, *SVM* is the Support Vector Machine, SGD is the stochastic gradient descent and KNN is the K-nearest neighbour. The numbers represent weighted accuracy / F1 measure where the evaluation is on the test set.

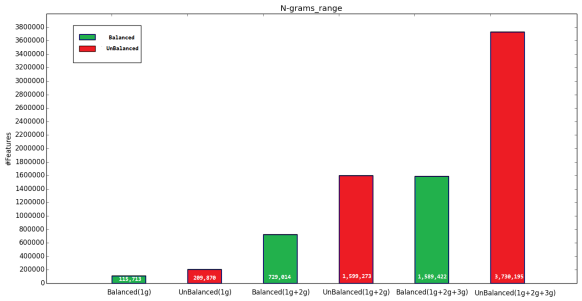

Figure VI.3: Number of uni grams, bigrams, and trigrams features per each class category

the test set where the performance measures are weighted F1 measure and total accuracy. Note that the inclusion of a third class "neutral" makes the problem much harder, and we get a lower performance than the case of two-class case ("positive" and "negative"). The reason is that there is a large confusion between neutral and positive, and between neutral and negative. Sometimes the numbered ratings (1 to 5), from which we extract the target class contradict what is written in the review, that even an experienced human analyzer will not get it right. Two accuracy measures are used to calculate the performance which are the total accuracyVI.2 and weighted F1 measureVI.3.

$$Tf\,Idf_{word,document} = log(1 + Freq(word, document)) * log(\frac{TotalDocuments}{TotalFreq(word)}) \quad \text{(VI.1)}$$

$$TotalAccuracy = \frac{Total\,number\,of\,true\,predicted\,reviews}{Total\,number\,of\,reviews} \quad \text{(VI.2)}$$

$$Weighted\,F1 = \frac{1}{N} * \sum_{i=1}^{i=N} F1(tag(i)) \quad \text{(VI.3)}$$

$$F1(tag(i)) = 2 * \frac{Precision(tag(i)) * Recall(tag(i))}{Precision(tag(i)) + Recall(tag(i))} \quad \text{(VI.4)}$$

The classifiers used in this experiment are widely used in the area of sentiment analysis and can be considered as a baseline for any further experiments. The classifiers used are:

1) **Multinomial Naive Bayes:** A well known method that is used in most NLP tasks. In this method each review is represented as a bag of words $\bar{X} =< x_1, x_2..., x_n >$ where the feature values are the term frequencies then the Bayes rule can be applied to form a linear classifier.

$$log(\,p(class|\bar{X})\,) = log(\,p(class)\,) * \prod_{i=1}^{n} p(\,x_i|class\,))$$

$$log(\,p(class|\bar{X})\,) = log(\,p(class)\,) + \sum_{i=1}^{n} log(p(\,x_i|class\,))$$

2) **Bernoulli Naive Bayes:** In this model features are independent binary variables that describe the input $\bar{X} =< 1, 0, 1...1 >$, which means the binary term occurrence is used instead of the frequency of the term in the bag of words model. the likelihood of a review given a class c is:

$$p(class|\bar{X}) = \prod_{i=1}^{n} [\,p(token_i|class)^{x_i} * (1 - p(token_i|class))^{1-x_i}\,]$$

3) **Support Vector Machine:** Linear SVM is a classifier that partitions the data using the linear formula $y = \bar{W} . \bar{X} + p$, selected in such a way that it maximizes

the margin of separation between the decision boundary and the class patterns (hence the name large margin classifier). SVM can be generalized to multiclass case using one versus all classification trick.

4) **Passive Aggressive:** It is an online learning model that uses a hinge-loss function together with an aggressiveness parameter C, in order to achieve a postive margin high confidence classifier. The algorithm is described in details in [**?**] with two alternative modifications that make the algorithm's ability to cope with noise better.
5) **Stochastic Gradient Descent:** It is an algorithm that is used to train other machine learning algorithms such as SVM where it samples a subset of training examples at every learning step. Then it calculates the gradient from this subset only, and uses the gradient to update the weight vector w in case of SVM classifier. It is widely used in case of large-scale machine learning problems[**?**].
6) **Logistic Regression:** The logistic regression uses a sigmoid function $h_w(x) = f(x) = \frac{1}{1+e^{-w^T x}}$ as a learning model, then it optimizes a cost function that measures the likelihood of the data given the classifier's class probability estimates. The cost function can be formulated as

$$Cost(\bar{w}) = \frac{-1}{m}\sum_{i=1}^{n}[y^{(i)}log(h_w(x^{(i)})) + (1-y^{(i)})log(1-h_w(x^{(i)}))]$$

where m is the total nuber of patterns, $x^{(i)}$is the pattern i and $y^{(i)}$ is the correct class of the pattern i.

7) **Perceptron:** It is a simple feed-forward single layer linear neural network with a unit step function as an activation function. It uses an iterative algorithm for training the weights, but that algorithm does not take into account the margin..
8) **K-Nearest Neighbour:** A simple well-known machine learning classifier based on distances between the patterns in the feature space. Specifically, a pattern is classified according to the majority class of its K-nearest neighbours.

From table V we can make the following observations:

1) The balanced set is more challenging than the unbalanced set. We think this is due to the fact it contains much fewer reviews compared to the unbalanced set, which makes it difficult for the classifiers to find the right decision boundary.
2) We can get an good overall accuracy and F1 of over 70% using especially SVM and logistic regression classifiers. This is consistent with previous results in **?** suggesting that SVM and Logistic Regression are a reliable choice.
3) Naive Bayes, Passive Agressive, and Perceptron are also a good choice of classifiers, with the careful choice of parameters.

Positive	Compound Positive	Negative	Compund Negative
رائع	يستحق القراءة	معجبنيش	لا يمكنني التعليق
ممتع	أتخيلني هناك	ممل	حار ونار
جميلة	أحسست الرواية	سيئة	توقعتها اجمل
سهل	أرفع قبعتي	ندمت	لم يروق

Table VI: Examples from the sentiment lexicon

VII. Sentiment Lexicon Generation

Manually constructing a sentiment lexicon is a formidable task due to the coverage issues and the whole process error prone. Also the problem of compound phrases appears while creating a manual sentiment lexicon. So we propose a simple method for extracting a baseline sentiment lexicon from LABR dataset this lexicon can be extended easily later. Our method utilize a beneficial features of the linear SVM and logistic regression as they inherently apply some sort of feature selection. This is because the weight values are an indication of the importance of the ngram. For example, ngrams that have negligible corresponding weights are deemed unimportant or ineffective. This is especially true if we use the L1 error measure for training the SVM (defined as norm $||x||_1 = \sum_i |x_i|$). In such case, many insignificant weight will end up being zero. So, we utilize this aspect to perform an automatic generation for the most informative ngrams by ordering the weights from SVM and logistic regression classifiers then select the highest 1000 weights as indiction for postive sentiment ngrams and the lowest 1000 weights as indiction for negative sentiment ngrams then we manually process them to remove any erroneous ngram. We end up with a list of 348 negative ngrams and 319 of positive ngrams also we constructed a list of 31 Arabic negation operators. Table VI gives some examples from the sentiment lexicon it is clear that some difficult compound phrases were captured using our approach.

VIII. Conclusion and Future Work

In this work we presented LABR dataset the largest Arabic sentiment analysis dataset to-date. We explored its properties and statistics, provided standard splits, and performed a comprehensive study, involving testing a wide range of classifiers and exporing their effects, Also we presentes a small sentiment lexicon that is extracted from the dataset. Our planned next steps include:

1) Expanding the sentiment lexicon and explore its potential.
2) Exploring using Arabic-specific and more powerful specially tailored features.

References