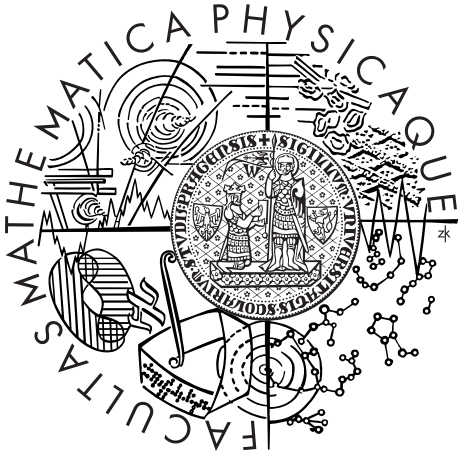
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Franky

Methods for Creating Subjectivity Lexicon for Indonesian

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: RNDr.Ondřej Bojar, Ph.D.
Mgr. Kateřina Veselovská

Study programme: Informatics

Specialization: Mathematical Linguistics

Prague 2013

I would like to thank those who made this thesis to happen.

To my wife who gives more than what I am worth for.
To my supervisors, Ondřej Bojar and Kateřina Veselovská, for their invaluable input, help, and guidance.
To my friends, for the data, tools, and entertainment.
To my family, for their distant supervision and their near love.
And to those who have helped me in my study, the coordinators, and all the administrative supports.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Název práce: Methods for Creating Subjectivity Lexicon for Indonesian
Autor: Franky
Katedra: Institute of Formal and Applied Linguistics
Vedoucí diplomové práce: RNDr. Ondřej Bojar, Ph.D., Mgr. Kateřina Veselovská
Abstrakt:
Cílem naší práce byla tvorba slovníků subjektivity pro indonéštinu překladem již existujících anglických slovníků a jejich následnou kombinací průnikem a sjednocením. Porovnali jsme efektivnost výsledných slovníků pomocí jednoduché prediktivní metody, která měří a porovnává počet výskytů kladných a záporných výrazů ve větě. Použili jsme také dvě různé hodnotící funkce založené jak na četnosti tak na relativní četnosti výrazů v neanotovaných datech. Úpravou prediktivní metody využívající strojového učení jsme posléze lépe začlenili údaje, které nemohly být zachyceny jednoduchou predikcí. Dále jsme ukázali, že slovníky byly v predikci na testovacích větách schopny dosáhnout vysokého pokrytí ale nízké přesnosti. Ohodnocování výrazů dokáže zlepšit pokrytí či přesnost, ovšem však vždy i se srovnatelným poklesem v druhé metrice. Predikce na bázi strojového učení byla schopna minimalizovat citlivost výkonu na velikosti slovníku, bude však zapotřebí dalších experimentů, aby se nalezla nejlepší volba prediktivní metody.
Klíčová slova: komputační lingvistika, postojová analýza, slovník hodnotících výrazů, indonéština

Title: Methods for Creating Subjectivity Lexicon for Indonesian
Author: Franky
Department: Institute of Formal and Applied Linguistics
Supervisor: RNDr. Ondřej Bojar, Ph.D., Mgr. Kateřina Veselovská
Abstract:
In this work, we created subjectivity lexicons of positive and negative expressions for Indonesian language by automatically translating English lexicons, and by intersecting and unioning the translation results. We compared the performances of the resulting lexicons using a simple prediction method that compares the number of occurrences of positive and negative expressions in a sentence. We also experimented with weighting the expressions by their frequency and relative frequency in unannotated data. A modification in prediction method using machine learning was later used to better incorporate the information that cannot be captured by the simple prediction. We showed that the lexicons were able to reach high recall but low precision when predicting whether a sentence is evaluative (positive or negative) or not (neutral). Scoring the expressions improve the recall or precision but with comparable decrease in the other measure. The machine learning prediction was able to minimize the sensitivity of the performances to the size of the lexicon, but further experiments are required to explore the best choice for the prediction method.
Keywords: computational linguistics, sentiment analysis, sentiment lexicon, subjectivity lexicon, Indonesian

Contents

Introduction

We present our work in the area of sentiment analysis for Indonesian language. In particular, we focus on building subjectivity lexicons using available resources in Indonesian. In this section, we give a brief overview and motivation for sentiment analysis and subjectivity lexicon, together with the description of the targets of our work.

Sentiment Analysis

Language, as one of the ways to communicate, can be seen as the tool to express individual subjective valuation that may involve the process of thinking, feeling, or a mixture of both. This expression can take the form of appraisal, opinion, or evaluation, that is expressed towards entities. Sentiment analysis is the study of how to analyze such expressions, that may include, but are not limited to, detection, extraction, and classification, which may be further processed to produce useful insight into the target entities.

The increasing interest in the field of sentiment analysis is mostly related to its practical applications in various settings. An example includes the use of sentiment analysis in commercial setting, e.g., in automatic summarization of customer's satisfaction over company products by analyzing their comments or feedback through social media, forums, or news. The other example is the use of sentiment analysis in political setting, in which we might get a summary of the popularity of a certain candidate. In Indonesia, the intensive use of Internet, especially social media, to express opinion or view on certain matter, marks the opportunity to further develop the research in this field.

Subjectivity Lexicon

While expressing a subjective view over an entity, we generally use words that convey certain senses or meaning that describe our thinkings or feelings toward the entity. In the simplest case, we can categorize these words as having positive or negative polarities. Positive, in the sense that they give a positive evaluation over the entity, and negative for negative evaluation. In English, the words 'good', 'wonderful', 'fantastic' can be marked as having positive polarity, while the words 'bad', 'awful' can be marked as having negative polarity. It is then both reasonable and profitable to possess a list that contains these words together with their polarities. We define subjectivity lexicon as the list that contains these words together with their positive and negative polarity information.

Having a subjectivity lexicon does not mean solving the problem of sentiment analysis. As described in [19], there are some aspects that need to be considered. For example, in a sentence like 'butuh ponsel yang murah meriah?' (Need a cheap good phone?) which is an interrogative sentence, the occurrences of words 'cheap' and 'good', which we can consider as positive aspects of a phone, do not necessarily mean that the sentence gives a positive evaluation to the 'phone'.

In the simplest case, a subjectivity lexicon can be just a list of positive and

negative words as in the list by [14]. Another possibility is to enrich the lexicon with additional information such as the intensity (weight) of the positivity and negativity or part-of-speech information, as in SentiWordNet ([9]).

We can also categorize a subjectivity lexicon as a general-purpose one or a domain-dependent or context-dependent one ([19]). A domain-dependent or context-dependent lexicon is a lexicon that is targeted to a specific domain or context. For example, the word 'small' in the phrase like 'small living room' when talking about property and phrase 'small risk' when talking about investment has different polarity that depends on the domain being discussed. Some works in this category are by [16] who address the domain-oriented sentiment analysis in Japanese and [7] who use Integer Linear Programming approach to adapt subjectivity lexicon into a certain domain.

A general-purpose lexicon, on the contrary, is not targeted to any specific domain. Some examples of the lexicons of this type are MPQA[1] and SentiWordNet[2]. A general-purpose lexicon focuses on collecting words that are generally acceptable by human as being positive or negative.

In this work, we experiment with subjectivity lexicons that are simple and only contains words and their polarity, and also general, not targeted to any specific domain.

Problem Definition

To our knowledge, there are no publicly available subjectivity lexicons for Indonesian. Some of the recent works that can be found in local publications as in [27] and [22] deal directly with the applicative aspect of sentiment analysis. Our goal is to provide a general subjectivity lexicon for Indonesian. There are a number of approaches to produce such lexicons from various resources, e.g., search engine, dictionaries, WordNet, corpora, etc (Chapter 1). However, these type of resources might not all be available for Indonesian.

Objective

The aim of this work is to explore approaches to produce subjectivity lexicons in Indonesian language, to test the resulting lexicons, and to compare their performances in sentiment prediction task. We would like to explore and implement, given the resources that we have, approaches that are usable, and hopefully, the most effective in producing the lexicons. Of the many possible approaches, in this work, we adapt the existing English subjectivity lexicons to Indonesian, mainly through translation using translation services, dictionaries, or building our own translation system. Other approaches to improve and extend the resulting lexicons are also considered.

[1] http://mpqa.cs.pitt.edu/

[2] http://sentiwordnet.isti.cnr.it/

Structure of the Thesis

We structure the presentation of our work as follows:

- Chapter 1 presents related work in subjectivity lexicon creation for other languages.
- Chapter 2 shows in detail the implementation and experiments that we performed to generate the lexicons, together with the evaluations, analysis, and discussion of the results.
- Chapter 3 describes our attempt to weight the expressions inside the lexicons.
- Chapter 4 shows the experiments of using machine learning in the sentiment prediction.
- Conclusion provides concluding remarks, suggestions, and some possibilities for future work.

1. Related Work

In this chapter, we present several works that are related to the creation of subjectivity lexicons. We divide the discussion of the works into subjectivity lexicon creation for English and non-English languages. This distinction is to differentiate the works between the language with a large number of language resources and the languages with probably fewer language resources.

1.1 English Subjectivity Lexicons

One of the earliest accounts that is related to the collection of words with polarity is the work in [13]. In this work, they experimented with adjectives that have semantic orientations or polarities. The purpose of the work is to identify and validate conjunction constraints used with respect to the polarity of the adjectives they conjoin. Despite the purpose, the interesting aspect of the work is that they collected and manually labelled 1,336 adjectives according to their semantic orientations, as positive or negative. This collection of adjectives may be seen as a subjectivity lexicon. The inter-reviewer agreement on the polarity of 500 adjectives taken from this list is high with average inter-reviewer agreement of 96.97%, which means that there are words that are correlated with certain polarity and that they are agreed among humans. The work also shows that the majority of adjectives in conjoined form (e.g., 'fair *and* legitimate', 'corrupt *and* brutal') tends to have the same polarity and that the morphological relationship between words, e.g., 'adequate' and '*in*adequate', can be used to identify the semantic orientation of the words. These two things may be useful if we want to extract and identify the polarity of words using the conjunction or morphological properties.

The idea that words or phrases contain certain polarities is also exploited in the work by [29]. In his work, Turney extracts phrases that consist of two words from review documents that follow a certain part-of-speech pattern. The semantic orientation (SO) of each phrase is determined by calculating the difference between the PMI (pointwise mutual information) of the phrase with the word 'excellent' and the word 'poor'.

$$PMI(w_1, w_2) = log_2 \left[\frac{p(w_1)p(w_2)}{p(w_1, w_2)} \right] \tag{1.1}$$

$$SO(phrase) = PMI(phrase,' excellent') - PMI(phrase,' poor') \tag{1.2}$$

simplified as:

$$SO(phrase) = log_2 \left[\frac{hits(phraseNEAR`excellent')hits('poor')}{hits(phraseNEAR`poor')hits('excellent')} \right] \tag{1.3}$$

The information of the count of the co-occurence (*hits*) of the phrase with the word 'excellent' or 'poor' is acquired by querying the search engine AltaVista that provided the operator 'NEAR', to constrain the search to documents that contain the phrase and the specified word within ten words of each other. A

positive score of SO, hence positive polarity word, indicates that the phrase is more closely related to the word 'excellent' and a negative score of SO indicates that it is more closely related to the word 'poor'. Using the average semantic orientation scores of phrases to classify reviews as positive or negative in several different domains, Turney gets accuracies ranging from 66% to 84%.

Currently available English subjectivity lexicons are created using several different approaches. Bing Liu's lexicon[1] is initially built from some manually created list of adjectives that are domain independent ([14]). The polarities of the words are manually identified. The list is expanded by adding adjectives that are located in the same sentence with the previously identified frequent product features from the review documents. The added adjectives are classified as positive or negative by identifying their antonymy or synonymy relationship with the words that have been identified previously. This relationship information is taken from the WordNet ([21]).

In the work by [31], a subjectivity lexicon containing over 8,000 words is used, called subjectivity clues. The lexicon is initially built by collecting subjective clues from several manually developed resources: [18], [4], [2], [13], [30]. Other words are taken from corpora and from subjective nouns learned from unannotated data [25]. The subjective nouns are extracted using some seed words to bootstrap the process of finding other words that fall into the same semantic category. They utilize Meta-Bootstrapping ([24]) and Basilisk ([28]) to search for patterns of words where the current seed words are located. The patterns and words are expanded iteratively. Some scoring mechanisms are used to ensure that the extracted words are in the same semantic category as the seed words. The initially produced lexicon is further expanded using the dictionary, thesaurus, and words from General Inquirer[2]. The positive, negative, both, or neutral tags are then assigned manually to the words whose polarities have not been identified yet.

Another subjectivity lexicon called SentiWordNet[3] is created on the basis of WordNet synsets. The lexicon contains WordNet synsets together with their positivity, negativity, and objectivity scores. The latest version, SentiWordNet 3.0 ([1]), is built in two steps: a semi-supervised learning and a random-walk.

In semi-supervised learning ([9]), three initial sets of positive, negative, and objective synsets are picked manually. Positive and negative sets are expanded using the relations defined in WordNet. The new synsets obtained by traversing relations that are considered to generate same polarity (e.g., also-see) are added to the same set where the new synsets are traversed from. Conversely, the synsets obtained by traversing relations that are considered to generate opposing polarity (e.g., antonymy) are added to their opposing polarity set. The objective set contains synsets that are not in positive or negative set. Each synset is then represented as a vector of synsets, by looking at the synsets in the gloss (definition) part. The Princeton WordNet Gloss Corpus[4] is used for this purpose. Two learners (SVM and Rocchio) are used to train ternary (positive, negative, objective) classifiers based on this training data. Each learner is trained using $k = 0, 2, 4, 6,$

[1] `http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html`

[2] `http://www.wjh.harvard.edu/inquirer/spreadsheet_guide.htm`

[3] `http://sentiwordnet.isti.cnr.it`

[4] `http://wordnet.princeton.edu/glosstag.shtml`

where k is the limit of the distance (number of iterations) between a synset in a positive or negative set with the newly traversed one, producing 8 different ternary classifiers. The resulting classifiers are applied to all WordNet synsets, and the positive, negative, and objective score of each synset is calculated as the average value across these 8 classifiers.

The second step of random-walk model used in SentiWordNet 3.0 is referred to as *inverse flow model* ([11],[10]). The input to the algorithm is an NxN matrix $\mathbf{W}$, where N is the total number of synsets, and $\mathbf{W}[i,j] = 1$ iff there is a link from synset n_i to synset n_j, based on whether synset n_j occurs in the gloss of synset n_i. The value of $\mathbf{W}[i,j] = 1$ is normalized to be $\mathbf{W} = \frac{1}{|F(i)|}$, where $F(i) = \{n_j|\mathbf{W}[i,j] = 1\}$. The output of the algorithm is a vector $\mathbf{a}$=$(a_1, a_2, ..., a_N)$, where each a_i represents the score of polarity of the synset. The value of $\mathbf{a}$ is calculated iteratively until convergence using the formula:

$$\mathbf{a^k} = \alpha\mathbf{a^{k-1}}\mathbf{W}^T + (1-\alpha)\mathbf{e} \tag{1.4}$$

The $0 \leq \alpha \leq 1$ is the control parameter and the $\mathbf{e}$ are defined so that $\sum_{i=1,2,..,N} e_i = 1$. The $\mathbf{e}$ is considered to control or contain the internal information of positivity or negativity that will be propagated from some selected synsets. Hence, the algorithm is run twice to produce positive scores by giving the null e_i for all synsets except for some selected positive synsets, and to produce negative scores by giving the null e_i for all synsets except for some selected negative synsets. The final scores of positivity, negativity, and objectivity are adjusted to sum up to 1 and are calculated as $score_{pos}(n_i) = x_1 a_i^{y_1}$, $score_{neg}(n_i) = x_2 a_i^{y_2}$, and $score_{obj}(n_i) = 1 - (score_{pos} + score_{neg})$, where x_1, y_1 (and x_2, y_2) are selected so that the distribution of the scores matches the original distribution before the adjustment.

1.2 Non-English Subjectivity Lexicons

For other languages, the creation of subjectivity lexicons that take the advantage of the availability of WordNet in their languages can be found in [3] and [23]. The work in [3] is for Hindi. They start with small seeds of 45 adjectives and 75 adverbs, pre-annotated with positive, negative, or objective polarity information. The seeds are expanded using Breadth First expansion by looking at the antonymy relation for opposite polarity and synonymy for the same polarity. In [23], the work to create Spanish subjectivity lexicon takes the advantage of aligned synsets between WordNets of different languages to do the mapping. They get two different lexicons. A full strength lexicon is created by taking words with strong negative or positive polarity from MPQA[5] lexicon and map them to the synsets in SentiWordNet, by taking the synset with the highest negative or positive value for each word. The found synsets are mapped to Spanish WordNet. The second lexicon, a medium strength lexicon, is created by mapping the synsets in SentiWordNet with polarity scores greater than 0.5 to Spanish WordNet.

A subjectivity lexicon creation for Dutch adjectives is performed in [26]. The approach is a mixture of manual annotation and automatic expansion. The first

[5] `http://mpqa.cs.pitt.edu/`

step is to extract adjectives with high frequencies from a collection of book reviews. Seven human annotators annotate the adjectives that are previously disambiguated using CORNETTO (an extension of Dutch WordNet). Each adjective is expanded by their best nearest neighbours (handpicked by two annotators) from the list of new adjectives taken from the corpus and using cosine similarity as the measure of similarity. Each adjective is represented as a vector of top 2,500 nouns from the same corpus. Another expansion is performed by adding words from the same synset in CORNETTO, and by using the relations provided, e.g., antonymy, synonymy.

A method of creating a subjectivity lexicon for a language with scarce resources (Romanian) is introduced in [5]. They propose a method to create subjectivity lexicon using an online dictionary and a collection of documents. The work uses a set of subjective words called seed words to bootstrap the lexicon creation. The process runs by querying the online dictionary using these seed words. A list of extracted words returned by dictionary for each seed word will then be filtered by calculating their similarity with the seed word using Latent Semantic Analysis (LSA). The LSA module is trained on half-million words Romanian corpus. The surviving words are added to the lexicon and the process continues until the maximum number of iterations is reached.

Some other approaches in subjectivity lexicon creation for non-English languages that do not utilize dictionary or thesaurus (e.g., WordNet) can be found in [20] and [15]. Their works can be considered as corpus-based approaches to lexicon creation. In [20], the authors use an underlying assumption that different types of corpus posit different characteristics of subjectivity or objectivity information. They use three different corpora of Wikipedia articles, news, and comments inside the news to build Dutch subjectivity lexicon. They take words in the news and comments that are not over-used in Wikipedia articles as subjective words. The measures of over-usage of words between the corpus are calculated using log-likelihood ratio and a DIFF calculation ([12]).

In [15], the authors exploit the dependencies and language structures in Japanese to extract polar sentences from a collection of one billion HTML documents. They use a list of cue words to detect the presence of polar clauses (positive or negative) in the dependency structure of the sentence. They also use layout structures, e.g., itemization/table, in HTML documents, and cue words such as 'pros and cons' and 'plus and minus' to extract positive and negative polar sentences. From the polar sentences, they extract candidate phrases consisting of adjectives and adjective phrases, e.g., noun+adjective, together with their counts in positive and negative sentences. The candidates are then filtered using chi-square and PMI polarity score, and pre-defined thresholds.

2. Building Subjectivity Lexicons by Translation

In this chapter, we describe the experiments that we performed on the Indonesian subjectivity lexicons created by translating English lexicons. We first describe the resources that we managed to collect. We then present the lexicons creation, evaluation procedures, and evaluation results of the lexicons.

2.1 Implementation Details

The major implementation of the work was performed in Python (v2.6/2.7) with additional libraries that include NLTK[1] (Natural Language Toolkit [6]) for sentence and word tokenization, and *html2text*[2] for the processing of HTML files.

2.2 Resources

In this section, we list the resources that we collected for our experiments.

2.2.1 Reviews

We collected reviews from `http://www.kitareview.com` that consist of reviews in several domains, e.g., books, restaurants, hardware. We managed to get around 900 reviews in total. We selected 876 reviews to be used as training data (unannotated), and 24 reviews to be annotated and used as a development/test set. The distributions of the training and the annotated data by their domains are shown in Table 2.1. We extracted the sentences from the reviews resulting in 14,998 training sentences and 446 development/test sentences. As can be seen from the table, the reviews from book, film, and hardware domains dominate the data.

Table 2.1: Domain Distribution on Selected Reviews

Domain	**Training (# Reviews)**	**Annotated (# Reviews)**
Book	284	4
Film	230	4
Game	57	2
Hardware	217	6
Music	1	2
Restaurant	69	2
Travel	5	2
Website	2	2

[1] `http://www.nltk.org`
[2] `https://github.com/aaronsw/html2text`

2.2.2 Annotated Sentences

We asked two annotators to annotate the 446 development/test sentences, marking the sentences as positive, negative, or neutral. We also asked the annotators to mark the expressions inside the sentences that they considered as having positive or negative polarities, and to add a flag 'TWOTARG' to the sentence that they considered as referring to more than one target, e.g., commenting on a picture and also on the main character of a book inside one sentence. The summary of the annotation result is shown in Table 2.2.

Table 2.2: Summary of Annotated Sentences

	Annotator 1	Annotator 2
Neutral Sentences	267	281
Evaluative Sentences	179	165
TWOTARG Sentences	30	17
# Pos Expressions (unique)	151	114
# Neg Expressions (unique)	40	33

We calculated the interrater agreements of the annotation result, using the Kappa (κ) statistic, on two different levels. The first one was calculated for all sentences where we calculated the agreement of annotating the sentences as neutral or evaluative. The second one was calculated only on the sentences where the two annotators agreed as evaluative, i.e., to get the agreement on positive and negative annotations on agreed evaluative sentences. Table 2.3 and Table 2.4 show the κ score for each of the cases. The interrater agreement on the neutral and evaluative sentences is lower compared to the interrater agreement on the evaluative sentences only. This might be an indicator that to agree on whether an evaluative sentence is positive or negative is easier than to agree on whether the sentence is neutral or evaluative.

Table 2.3: Interrater Agreement on All 446 Sentences ($\kappa = 0.697$)

		Annotator 2	
		Neutral	Evaluative
Annotator 1	Neutral	242	25
	Evaluative	39	140

Table 2.4: Interrater Agreement on 140 Agreed Evaluative Sentences ($\kappa = 0.921$)

		Annotator 2	
		Positive	Negative
Annotator 1	Positive	125	1
	Negative	1	13

2.2.3 English Subjectivity Lexicons

We collected four different subjectivity lexicons for the purpose of our work. All of the lexicons are English subjectivity lexicons. We decided to use English lexicons

as our source lexicons because more of them exist than for other languages and most of them can be acquired easily. The other reason is that the resources needed to perform automatic translation from English to Indonesian are easier to collect, and some, such as online dictionaries, are readily available.

The four lexicons used in our work are listed below:

1. Bing Liu's Opinion Lexicon
   The lexicon is created and maintained by Bing Liu ([14]). It contains a list of positive and negative words for English without any additional information (it consists of around 6,800 words)[3].

2. Harvard General Inquirer
   General Inquirer lexicon[4] contains words together with the information about their syntactic and semantic categories. Among them are the positive and negative categories that can be used to extract the required positive and negative words.

3. MPQA (Multi-Perspective Question Answering) Subjectivity Lexicon
   MPQA[5] lexicon is built using manual and automatic identification of polar words and used in the work in [31] for phrase-level sentiment analysis. The lexicon contains words and information about their polarities, subjectivity strengths, and also their part-of-speech tags. It is the lexicon used by OpinionFinder system[6].

4. SentiWordNet
   SentiWordNet is a lexicon built by assigning positive and negative scores to WordNet synsets. The author uses a random walk model to propagate the initial positive and negative value using the link found in the gloss of the synsets. The final lexicon contains list of synsets together with their positive and negative scores.

We pre-processed the lexicons to extract the required positive and negative words (expressions) from each of the lexicons. For Bing Liu's lexicon, the expressions are already separated into two different list, positive and negative, and hence can be used directly. From General Inquirer lexicon, we extracted expressions that have value of 'Positiv' on the 'Positiv' column as positive expressions, and words that have value of 'Negativ' on the 'Negativ' column as negative expressions.

Positive and negative expressions from MPQA lexicon were extracted by taking the words from tag 'word1'. The polarity of the extracted words can be identified from the tag 'priorpolarity'. We only took words with 'priorpolarity' value of 'positive', 'negative', or 'both'. In the case of 'both', we put the word in both positive and negative lexicons.

SentiWordNet stores the expressions as a list of synsets. Each synset has positive and negative scores, ranging from 0 to 1. We extracted the words from each synset as positive expressions or as negative expressions if it had positive or

[3] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
[4] http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm
[5] http://mpqa.cs.pitt.edu/
[6] http://mpqa.cs.pitt.edu/opinionfinder/

negative scores greater than or equal to 0.5 (>= 0.5). The numbers of positive and negative expressions extracted from each lexicon are shown in Table 2.5[7].

Table 2.5: Numbers of Expressions Extracted from English Subjectivity Lexicons

Lexicon	# Pos Expressions	# Neg Expressions
Bing Liu	2,006	4,783
General Inquirer	1,915	2,291
MPQA	2,321	4,168
SentiWordNet	5,730	8,821
Intersection	470	791
Union	7,809	12,445

SentiWordNet has the highest number of expressions for both positive and negative expressions. It might be caused by some specific terms like the name of disease, and the existence of complex words like 'reliableness' (positive), which cannot be found in the other lexicons. For all lexicons, the number of negative expressions is higher than the positive one.

The intersection shows the numbers of shared (agreed) positive and negative expressions by all of the lexicons. The shared expressions constitute around 10% (for SentiWordNet) to around 30% (for General Inquirer) of the total number of expressions of each lexicon. The union shows the total numbers of unique positive and negative expressions from all of the lexicons. From the numbers, we can infer that there are also expressions that are shared by some (not all) of the lexicons.

2.2.4 Indonesian-English Parallel Corpus

We built our parallel corpus by collecting documents that are written both in English and Indonesian from several sources: BBC[8], RBC Ministries[9], SMERU[10], and AusAID[11]. BBC provides news in English and Indonesian version. Documents from RBC Ministries are daily devotional articles written in several languages including English and Indonesian. SMERU produces reports of their researches about public policy. AusAID, the Australian Agency for International Development, provides reports about their activities in Indonesia. The sentences inside the documents were extracted and aligned manually. The numbers of sentences, word tokens, and word types (lowercased, punctuation removed) from each source are listed in Table 2.6.

Comparing the number of tokens with the number of sentences, we can estimate that the average sentence length is about 15-20 word tokens. The table also shows that the number of unique words (word types) is small compared to the total number of tokens. Hence, there might be a lot of same words used repetitively in the corpus.

[7]We found duplicated entries in our English General Inquirer lexicon. The numbers without duplication are 1,637 for positive lexicon and 2,005 for negative lexicon. This duplication should not affect the resulting Indonesian lexicons, since we run de-duplication process before producing the final lexicons.

[8]`http://bbc.co.uk/indonesia/topik/dwibahasa/`

[9]`http://odb.org`

[10]`http://www.smeru.or.id`

[11]`http://www.ausaid.gov.au`

Table 2.6: English-Indonesian Parallel Corpus

Source	# Sentences	# Word Tokens		# Word Types	
		Indonesian	English	Indonesian	English
BBC	468	7,640	8,462	2,277	2,520
AusAID	3,112	66,090	67,504	4,548	4,902
RBC Ministries	9,823	155,558	155,983	10,888	11,472
SMERU	26,966	518,976	606,403	13,358	13,703
Total	40,369	748,264	838,352	20,805	21,226

2.3 Building the Lexicons

We built our subjectivity lexicons for Indonesian by translating the English lexicons using several methods. We also tried to create new lexicons by applying union and intersection operations to the resulting translations.

2.3.1 Lexicon Structure

For a better flexibility of the prediction function, we set each lexicon to have two columns. The first column is the expressions (words) and the second column is the weight of the expression. Unless specified otherwise, the default weight of each expression is 1.0. This structure is to handle the weighting of lexicon entries as explained in Chapter 3.

2.3.2 Translating English Lexicons

The English lexicons were translated using three different methods, resulting in 12 different lexicons, with 4 lexicons for each translation method.

1. Google Translate[12]
   We used the web interface of Google Translate to translate each lexicon by copying and pasting all the words from the lexicon. The translations were done in November 2012.

2. Moses[13]
   We built a statistical machine translation from English to Indonesian using Moses and our parallel corpus that consists of 40,369 sentences with no additional annotation. We used 38,369 sentences for training the translation system and 2,000 sentences for tuning. The translation system was trained using default parameters.

   We tested the performances of the translation system by performing 3-fold cross validation. We created three different sets of sentences with each set contains 36,369 sentences for training, 2,000 sentences for testing, and 2,000 sentences for tuning. We randomized the sentences before creating the three data sets. The first fold contains the first 2,000 sentences as test sentences, and the next (second) 2,000 sentences for tuning. The rests were used as

[12] `http://translate.google.com`
[13] `http://www.statmt.org/moses`

training sentences. The second fold contains the second 2,000 sentences for testing, and the third 2,000 sentences for tuning, while the rests were used for training. The third fold contains the third 2,000 sentences for testing, and the fourth 2,000 sentences for tuning, and the rests were used for training. The average BLEU score of training and testing the translation systems (without tuning) on these three data sets is 31.36.

3. Kamus.net[14]
   We used Kamus.net as our online English-Indonesian dictionary. We created a script to translate each word and took the first translation of each expression.

The resulting translations were further processed by removing untranslated and duplicated words. Manual filtering performed by one person was performed to remove words that do not contain positive or negative polarity. Expression that consist of more than one word like 'keadaan yang menyedihkan' (a sad/an unfortunate condition) which does not constitute a multi-word expression was also removed. We kept the multi-word expressions that convey positive or negative polarity, e.g., 'acungan jempol' (thumbs up). We tried to maintain the consistency by ensuring that words removed or maintained in one type of lexicon are also removed or maintained in other lexicons. The resulting numbers of positive and negative expressions are shown in Table 2.7.

Table 2.7: Positive and Negative Expressions for Translation Method

Translation	Lexicon	Original		Remove Untranslated and Duplications		Final	
		Pos	Neg	Pos	Neg	Pos	Neg
Google	Bing Liu	2,006	4,783	1,147	2,589	740	1,500
	General Inq	1,915	2,291	1,203	1,443	690	911
	MPQA	2,321	4,168	1,429	2,426	796	1,359
	SentiWordNet	5,730	8,821	3,404	4,857	873	1,205
Moses	Bing Liu	305	302	249	255	180	165
	General Inq	606	382	379	245	237	130
	MPQA	433	322	372	277	236	158
	SentiWordNet	1,051	1,223	847	886	236	160
Online Dict	Bing Liu	1,268	2,914	641	1,290	478	910
	General Inq	1,567	1,871	884	1,009	536	692
	MPQA	1,565	2,553	887	1,271	560	871
	SentiWordNet	2,892	3,647	1,606	1,856	582	1,221

The column 'Original' shows the number of words after the translation process. Google Translate gives results for all words, with unknown words returned as the original words (untranslated), and hence, smaller number of words after the removal of untranslated words and de-duplication process. The final results

[14] `http://www.kamus.net`

have a smaller number of words since the words might be translated into non-evaluative words in Indonesian or sometimes into clauses which are not multi-word expressions. The SentiWordNet loses many of its words since it contains a lot of specific names of diseases, scientific names or terms, etc, that we considered as non-evaluative and too specific.

Moses produces a small number of words since the vocabularies from the training data might not be large enough and come from a different domain. Most of the words from the lexicons cannot be translated. The online dictionary produces more words. However, the resulting translations contain many repeated entries that causes the number of words to be significantly smaller after the deduplication process. All the lexicons, except the ones from Moses translation, still maintain the property of having more negative expressions than positive one.

2.3.3 Intersections and Unions

We produced new lexicons by performing intersection and union operations on the resulting lexicons. The operations were performed for each type of source and each translation method. The intersection operation was performed to get the agreed expressions from lexicons of different sources or different translation methods. The union operation was intended to collect all possible expressions. Since the resulting lexicons from Moses translation are small compared to the others, we decided to ignore the lexicons from Moses translation when performing intersection operations on the lexicons of the same source and when intersecting all lexicons. This was done to ensure that we do not get too small number of expressions. Table 2.8 shows the resulting lexicons after the intersection and union operations.

Table 2.8: Positive and Negative Expressions after Intersection and Union

Operation	Type	Description	Pos	Neg
Intersection	Translation Method	Google	364	551
		Moses	92	78
		Online Dict	306	448
	Source (w/o Moses)	Bing Liu	330	660
		General Inq	330	444
		MPQA	376	619
		SentiWordNet	388	543
	All (w/o Moses)	All	178	270
Union	Translation Method	Google	1256	1921
		Moses	366	246
		Online Dict	788	1565
	Source	Bing Liu	932	1781
		General Inq	963	1185
		MPQA	1040	1638
		SentiWordNet	1112	1918
	All	All	1557	2665

For the intersection of translation method, the portion of shared expressions of the lexicons under the same translation method is more than 50% of the lex-

icon with the smallest number of expressions. Looking at the intersection of lexicons from the same source, we can see that there is a significant drop in the number of negative expressions for SentiWordNet. We think that this is caused by the different translations provided by Google Translate and the online dictionary. The union operation, as expected, shows an increase in the number of expressions. The total number of unique words after unioning all lexicons is significantly smaller than the union of their corresponding English lexicons, but still relevant considering the smaller number of expressions each lexicon has. The numbers in the table also show that the lexicons have some differences in the expressions they have and might give different performances in prediction.

2.4 Evaluation Setup

The evaluation of the resulting lexicons was performed against the set of test sentences that have been annotated manually with their polarities, as positive, negative, or neutral sentences. The performance of a lexicon was evaluated with respect to how well they predict the true polarity of the sentences.

2.4.1 Simple Prediction Method

In order to make the prediction, we needed to decide what sentiment analysis method that we were going to use for the prediction task. Since our goal was to compare the lexicons, we considered that the choice of the method would be less important for now. As long as we consistently used the same sentiment analysis method when evaluating the lexicons, the results produced should be able to fairly compare the performances of these different lexicons. We decided to use a simple prediction method for the prediction. We also incorporated the weight of the expressions into the prediction method that would be useful when we assigned the weights in the next experiments. The polarity (sentiment) of a given sentence s that contains a list of positive expressions P and a list of negative expressions N is defined as:

$$polarity(\mathrm{s}) = \begin{cases} positive & \sum_{p \in P} weight_{pos}(\mathrm{p}) > \sum_{n \in N} weight_{neg}(\mathrm{n}) \\ negative & \sum_{p \in P} weight_{pos}(\mathrm{p}) < \sum_{n \in N} weight_{neg}(\mathrm{n}) \\ neutral & \sum_{p \in P} weight_{pos}(\mathrm{p}) = \sum_{n \in N} weight_{neg}(\mathrm{n}) \end{cases}$$

Several criteria of how we detected the expressions inside the sentences are:

- **Unique Polarity.** An expression in a sentence can only be tagged with one type of polarity, either as positive or as negative expression. In other words, no single expression will be included as both positive and negative.
- **Prioritize positive expressions.** We first search for the occurrences of positive expressions inside the sentence and then continue with the negative expressions.

- **Prioritize longer expressions.** Since there is a possibility that a shorter expression is a part of a longer expression, we collect the counts by first matching longer expressions.

- **Negation.** We adapt technique presented in [8] to handle the negation of sentiment caused by a negation word. Negation switches the polarity of an expression as in 'good' and 'not good'. We use words 'tidak', 'tak', 'tanpa', 'belum', and 'kurang' as negation words. The words that occur between the negation word and the first punctuation after the negation word will be tagged with 'NOT_', e.g. 'kurang bagus gambarnya ?' (the picture is not good enough ?) to 'kurang NOT_bagus NOT_gambarnya ?'.

2.4.2 Evaluation Measures

We measured the performances of the lexicons using accuracy, precision, recall, and F-measure.

1. Accuracy
   Accuracy measures the overall performance of predicting the correct polarity of the sentences, as positive, negative, or neutral sentences. The formula is given by:

$$Accuracy = \frac{c_{pos,pos} + c_{neg,neg} + c_{neu,neu}}{n} \tag{2.1}$$

   where:

$c_{a,b}$	:	count of predicted as b sentences with polarity a
pos	:	positive
neg	:	negative
neu	:	neutral
n	:	total number of test sentences

2. Precision, Recall, and F-Measure
   We measured precision, recall, and F-measure on four different categories: neutral, evaluative, positive, and negative. The evaluative means that the sentence is a non-neutral sentence. In calculation, an evaluative sentence is correctly predicted if it is predicted as non-neutral sentence, regardless of being predicted as positive or negative.

   The formulas for precisions are given by:

$$P\text{-}NON = \frac{c_{neu,neu}}{c_{pos,neu} + c_{neg,neu} + c_{neu,neu}} \tag{2.2}$$

$$P\text{-}POS = \frac{c_{pos,pos}}{c_{pos,pos} + c_{neg,pos} + c_{neu,pos}} \tag{2.3}$$

$$P\text{-}NEG = \frac{c_{neg,neg}}{c_{pos,neg} + c_{neg,neg} + c_{neu,neg}} \tag{2.4}$$

$$P\text{-}EVL = \frac{c_{pos,pos} + c_{neg,neg} + c_{pos,neg} + c_{neg,pos}}{c_{pos,pos} + c_{neg,neg} + c_{pos,neg} + c_{neg,pos} + c_{neu,pos} + c_{neu,neg}} \tag{2.5}$$

The formulas for recalls are given by:

$$\text{R-}NON = \frac{c_{neu,neu}}{c_{neu,pos} + c_{neu,neg} + c_{neu,neu}} \quad (2.6)$$

$$\text{R-}POS = \frac{c_{pos,pos}}{c_{pos,pos} + c_{pos,neg} + c_{pos,neu}} \quad (2.7)$$

$$\text{R-}NEG = \frac{c_{neg,neg}}{c_{neg,pos} + c_{neg,neg} + c_{neg,neu}} \quad (2.8)$$

$$\text{R-}EVL = \frac{c_{pos,pos} + c_{neg,neg} + c_{pos,neg} + c_{neg,pos}}{c_{pos,pos} + c_{pos,neg} + c_{pos,neu} + c_{neg,pos} + c_{neg,neg} + c_{neg,neu}} \quad (2.9)$$

We will use F1 score for the F-Measure so that to put the same importance on precision and recall values. The formulas are given by:

$$F1\text{-}NON = \frac{2 * \text{P-}NEU * \text{R-}NEU}{\text{P-}NEU + \text{R-}NEU} \quad (2.10)$$

$$F1\text{-}POS = \frac{2 * \text{P-}POS * \text{R-}POS}{\text{P-}POS + \text{R-}POS} \quad (2.11)$$

$$F1\text{-}NEG = \frac{2 * \text{P-}NEG * \text{R-}NEG}{\text{P-}NEG + \text{R-}NEG} \quad (2.12)$$

$$F1\text{-}EVL = \frac{2 * \text{P-}EVL * \text{R-}EVL}{\text{P-}EVL + \text{R-}EVL} \quad (2.13)$$

2.4.3 Test Data

Our test data consists of 380 sentences that were taken from the annotated data where the two annotators agree on their polarities. The test data consist of 125 positive, 13 negative, and 242 neutral sentences. Due to the small number of negative sentences, the resulting percentages of precision, recall, etc should be interpreted carefully, since it does not really represent one hundredth of some data size, but only one tenth of it.

2.4.4 Baselines

We created three different types of baselines for the experiment.

- **Oracle (ORACLE)**. The first baseline was created as the upper bound of the prediction, which can also be considered as the gold standard. We created a lexicon containing positive and negative words taken from the test data. The positive and negative words are the words that were tagged by the annotators when annotating the test data.

- **3-fold Cross Validation (BASE-CV)**. The second baseline was taken as the average of a 3-fold cross validation on the test data. The test data was randomly split into 3 set (fold) of sentences, and we named the sets as CV0, CV1, and CV2. From each fold, we took positive and negative words that were tagged previously by the annotators, as in the creation of Oracle, resulting in three list of positive and negative expressions, one for each fold.

The prediction on each fold was performed using the union of lexicons taken from the other two folds, and hence, for this purpose, we created three new lexicons: BASECV12, BASECV01, and BASECV02. The last two digits on the lexicon's name specify from which folds the lexicon was built. For example, BASECV12 is the collection of positive and negative words taken from CV1 and CV2.

- **Uniform Prediction**. The last baseline consists of three different prediction methods which did not use any lexicons to do the prediction. Each of the predictions was a simple prediction method, which was to always guess the sentences as positive (BASE-ALLPOS), negative (BASE-ALLNEG), or neutral (BASE-ALLNON).

The evaluation results for the baselines are shown in Table 2.9.

Table 2.9: Evaluation Results of the Baselines

Measure	ORACLE	BASE-CV	BASE-ALLPOS	BASE-ALLNEG	BASE-ALLNON
Accuracy	86.84%	75.77%	32.89%	3.42%	63.68%
P-EVL	79.49%	76.29%	36.32%	36.32%	0.00%
R-EVL	89.86%	54.49%	100.0%	100.0%	0.00%
F1-EVL	84.35%	63.45%	53.28%	53.28%	0.00%
P-NON	93.75%	77.54%	0.00%	0.00%	63.68%
R-NON	86.78%	89.99%	0.00%	0.00%	100.0%
F1-NON	90.31%	83.28%	0.00%	0.00%	100.0%
P-POS	82.09%	79.77%	32.89%	0.00%	0.00%
R-POS	88.00%	54.85%	100.0%	0.00%	0.00%
F1-POS	84.94%	64.62%	49.50%	0.00%	0.00%
P-NEG	45.45%	33.33%	0.00%	3.42%	0.00%
R-NEG	76.92%	13.33%	0.00%	100.0%	0.00%
F1-NEG	57.14%	19.05%	0.00%	6.62%	0.00%

The evaluation of the ORACLE returns high scores for almost all measures. An exception is in the precision of predicting the negative sentences which is below 50%, although the recall is quite high (10 out of 13 sentences). This might be caused by the small number of negative sentences in the test data, and hence, it is over-predicting some non-negative sentences as negative.

The BASE-CV still returns high scores for some of the measures. The difference with the ORACLE is in the significant drop on the recalls of the evaluative sentences. It means that some of the evaluative sentences are incorrectly predicted as non-evaluative. This might be due to the fact that some evaluative words in test fold are not present in the lexicon built from the other two folds.

The accuracies of BASE-ALLPOS, BASE-ALLNEG, and BASE-ALLNON reflect the distribution of the positive, negative, and neutral sentences inside the data. We can see from the table that when we predicted all sentences as neutral, we can still get an accuracy of 63.68 %, although all other measures related to the evaluative sentences will be 0%.

2.5 Evaluations

Given the evaluations measures presented in 2.4.2, we decided to simplify the analysis of the evaluations using only some of the measures. We give a focus on the evaluations of the lexicons with regards to predicting the evaluative sentences, which in this case is measured by P-EVL and R-EVL. The P-POS/R-POS and P-NEG/R-NEG are discussed if more detail is required. We show the accuracy as the measure of the overall performance of the lexicons. The results are shown mostly in a precision-recall graph, to easily compare the precision and recall performances of the lexicons.

2.5.1 Naming Convention of the Lexicons

The basic lexicons produced from translations are named by concatenating the translation method and the lexicon name, e.g., GTR_BING (Google Translate and Bing Liu lexicon). The names of the lexicons from intersection operations are marked with ITS_ and from union operations with UNI_, e.g., ITS_MOS is a lexicon produced from intersecting all lexicons translated using Moses. Another naming convention that is used is ATR (all translations/all lexicons), e.g., UNI_ATR is a lexicon produced from combining all the basic lexicons. Table 2.10 summarizes the naming.

Table 2.10: Naming Convention of the Lexicons

Name	Description
BING	Source lexicon is Bing Liu lexicon
GINQ	Source lexicon is General Inquirer lexicon
MPQA	Source lexicon is MPQA lexicon
STWN	Source lexicon is SentiWordNet lexicon
GTR	Translated using Google Translate
MOS	Translated using Moses
OEI	Translated using Online Dictionary
ITS	The result of intersection operation
UNI	The result of union operation
ATR	All basic lexicons

2.5.2 Evaluative and Neutral Sentences Prediction

Regarding the measures of P-NON and R-NON, we show in Figure 2.1 the relationship between the measures of P-EVL/R-EVL with the P-NON/R-NON. The points inside the circle are the results of evaluations in predicting the non-evaluative sentences (P-NON, R-NON). The points outside the circle are their counterparts, i.e., the results of evaluations in predicting evaluative sentences (P-EVL, R-EVL). The lines show some examples of the relationship of the evaluation results of the same lexicon used in these two different types of predictions.

The UNI_ATR in non-evaluative sentences prediction (NON) has a low recall and high precision, while its counterpart in evaluative sentences prediction (EVL) has a high recall and low precision. For ITS_MOS, we can see that it has high

recall and precision for non-evaluative sentences prediction but low recall and precision for evaluative sentences prediction. The GTR_MPQA for non-evaluative sentences prediction has a low recall and high precision but high recall and low precision in evaluative sentences prediction.

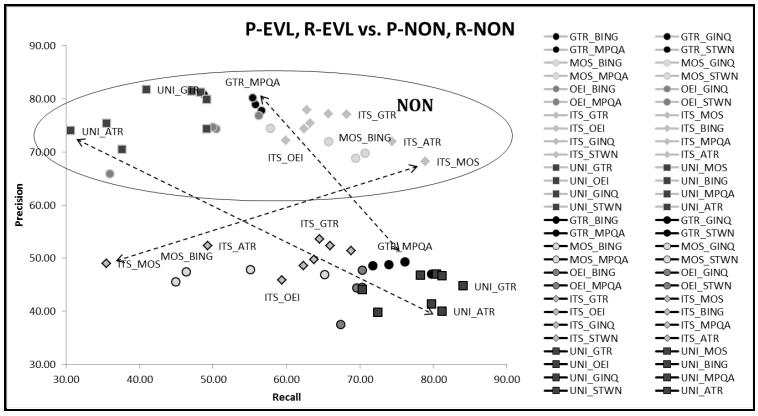


Figure 2.1: Comparison of P-EVL/R-EVL and P-NON/R-NON of All Lexicons

From the figure, we can somehow predict the behaviour or the precision and recall of the non-evaluative sentences prediction by looking at the precision and recall of the evaluative sentences prediction. Since they come from the same test data, the low precision and high recall on the evaluative sentences prediction mean that there are more sentences predicted as evaluative while they are actually non-evaluative sentences, and this corresponds to the low recall but high precision on the non-evaluative sentences prediction. Due to this property, we can look at the performances of evaluative sentences prediction to get a hint of the performances of non-evaluative sentences prediction.

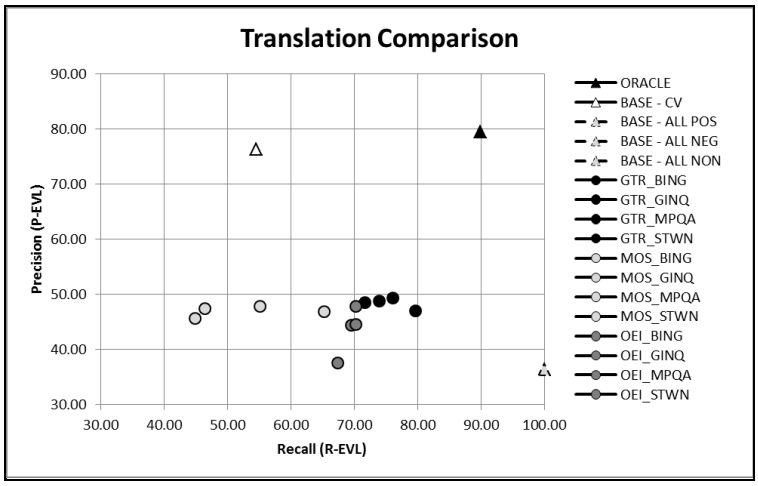


Figure 2.2: Precision-Recall of Basic Lexicons from Translation Point of View

2.5.3 Basic Lexicons

We compare the resulting lexicons from two different viewpoints, the translation method and the source lexicon. Figure 2.2 shows the comparison of the lexicon

from the point of view of translation method. As can be seen from the graph, the lexicons from Google Translate seem to have better recalls, while the lexicons from Moses translation have lower recalls compared to the others. Translation using the online dictionary produces lexicons that have recalls around 70%. In terms of precision, they do not have significant differences. The precisions seem to fall between 40% and 50% with the exception of one of the instance of the online dictionary translation that has the precision lower than 30%.

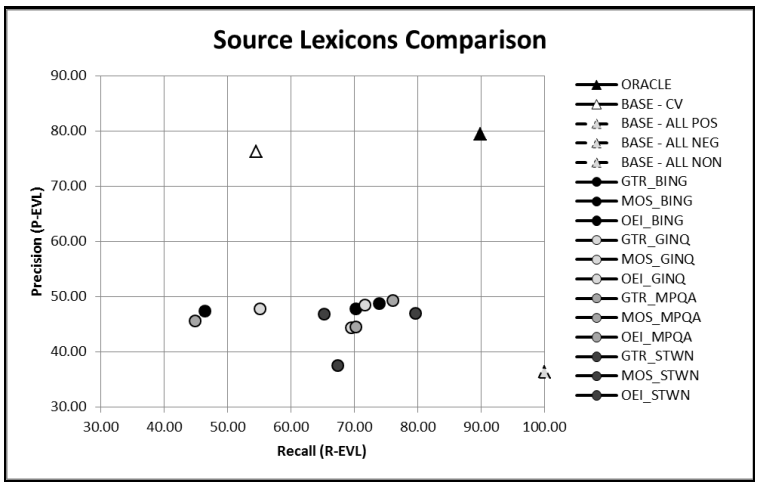


Figure 2.3: Precision-Recall of Basic Lexicons from Source Lexicon Point of View

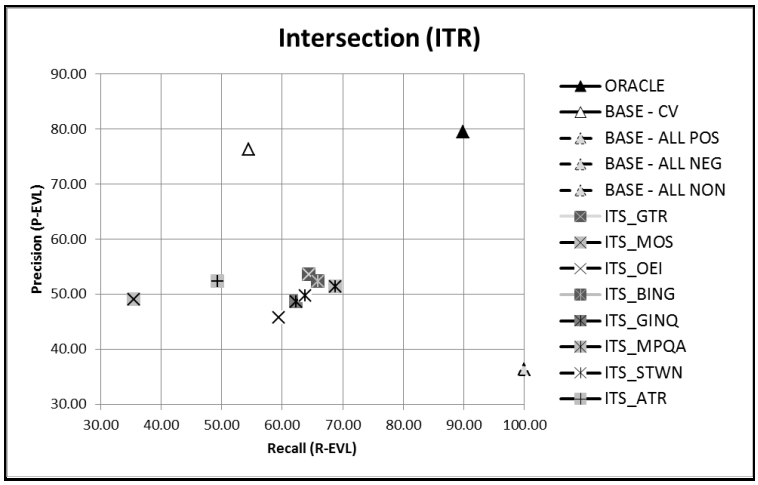


Figure 2.4: Precision-Recall of Intersection Operations

Figure 2.3 compares the lexicon from the point of view of source lexicon. The lexicons from SentiWordNet seems to have highest recalls most of the time but have lower precisions compared to the other lexicons that have steady performances on the precisions.

Comparing the lexicons with the baselines, we can see that they are better at precision compared to the BASE-ALLPOS and BASE-ALLNEG, although they have lower recalls. We see a striking difference between the performance of all the lexicons and the simple data-driven baseline (BASE-CV). While the recall of 54% is below the average, the precision is better and reaches 76%. This is very close to the upper bound we can hope for, i.e., the ORACLE, with P-EVL of 79%. Put

simply, the generic lexicons are perhaps good for high-recall applications but it will be very difficult for them to beat a 'custom' lexicon created from just a few dozens of in-domain sentences. The BASE-ALLNON is not shown on this graph since it has zero precision and recall in evaluative predictions.

2.5.4 Intersection and Union Lexicons

Figure 2.4 and Figure 2.5 show the evaluation results of the lexicons from intersection and union operations. The figures indicate that the resulting lexicons have the same performance as basic lexicons if compared to the baselines. As before, in most cases, they have lower recalls and higher precisions than BASE-ALLPOS and BASE-ALLNEG, lower precisions and higher recalls than BASE-CV, and lower recalls and precisions than the ORACLE. The operations performed seem to not able to give significant improvement in the precision and recall values.

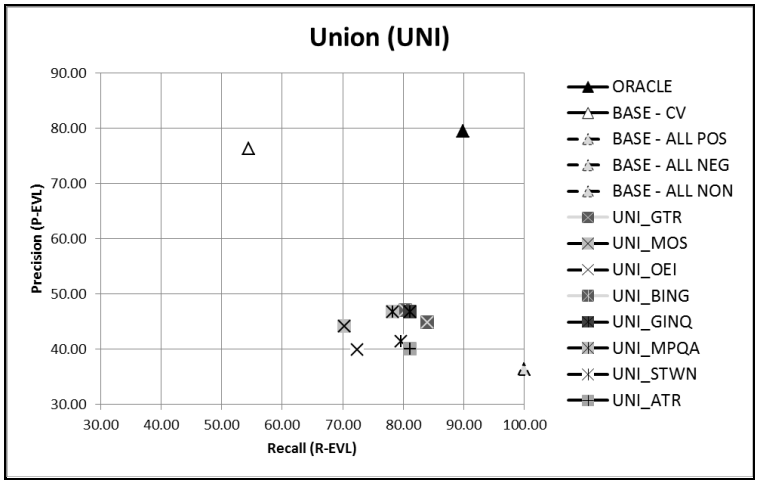


Figure 2.5: Precision-Recall of Union Operations

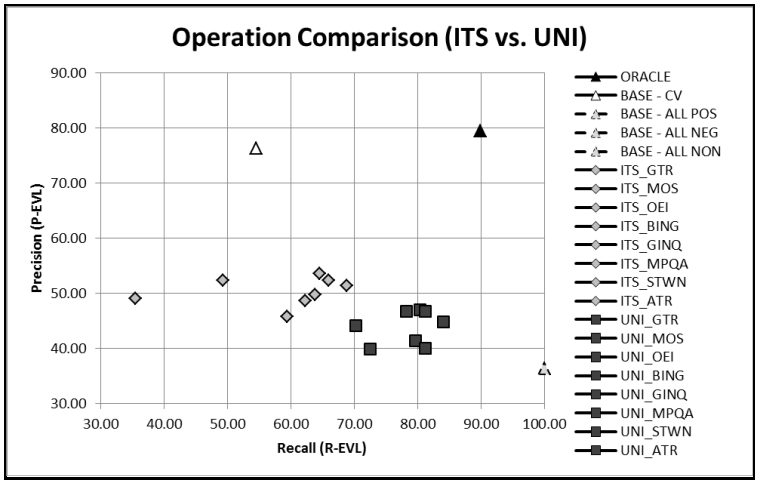


Figure 2.6: Comparison of Intersection and Union Operations

Looking at the individual operations, we can see that intersection operation is able to increase the precision to be greater than 50%, with the highest precision

reached by ITS_GTR. The recalls are in the range of 60%-70%. The exceptions are the lexicons from ITS_MOS and ITS_ATR that we think are caused by the small number of expressions in the lexicons. The ITS_ATR intersects all of the lexicons, while ITS_MOS intersects lexicons from Moses translation which originally have a small number of expressions. The union operation brings the lexicons into higher recalls and obviously slightly losses in the precisions.

Comparing the two operations in Figure 2.6, we can see how the lexicons from union operations are located in the high recall region, while the lexicons from intersection operations are located in a slightly higher precision region with lower recalls.

2.5.5 Comparing All Produced Lexicons

Plotting all the lexicons in Figure 2.7, we can see that the union operations produced lexicons with no significant improvement in the precision, but they are able to gain recalls in most cases. The intersection operations are able to improve the precisions to be above 50%, although the recalls are lower and stay under 70%.

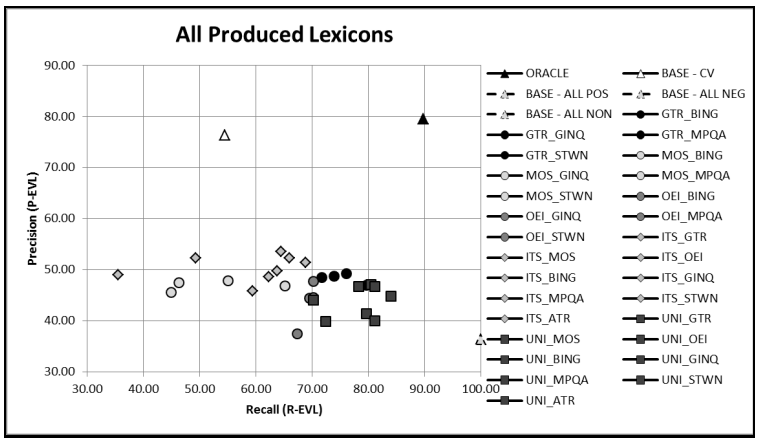


Figure 2.7: Comparing All Produced Lexicons

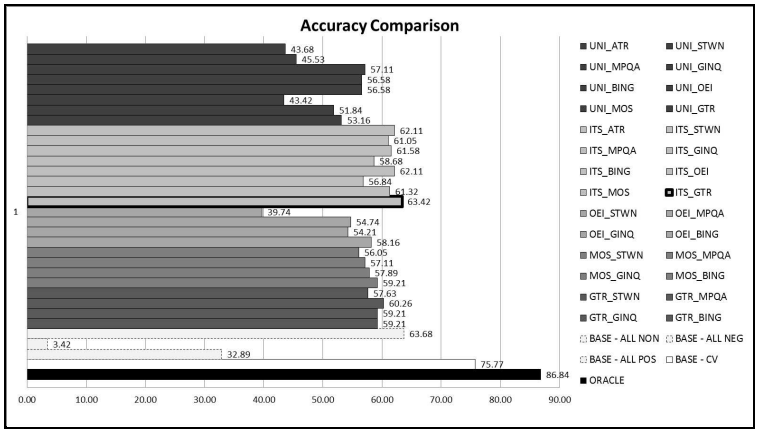


Figure 2.8: Accuracy Comparison of All Lexicons

We shows the comparison of the lexicons in accuracy, precision, and recall values in Figure 2.8, Figure 2.9, and Figure 2.10. The ITS_GTR lexicon has the

highest accuracy and precision compared to other produced lexicons. The highest recall is reached by the UNI_GTR lexicon.

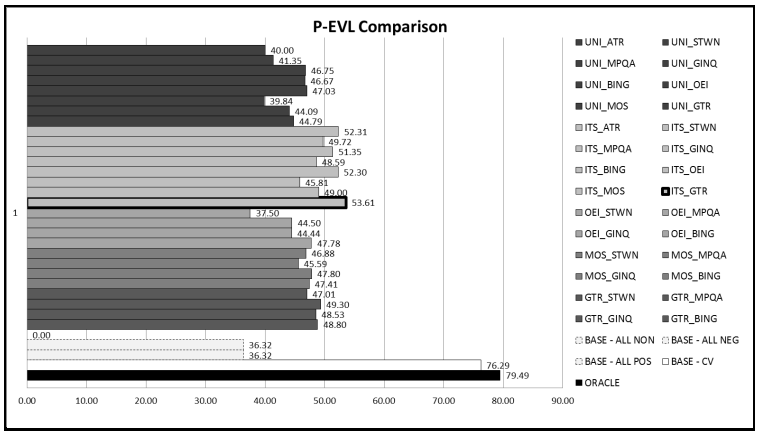


Figure 2.9: P-EVL Comparison of All Lexicons

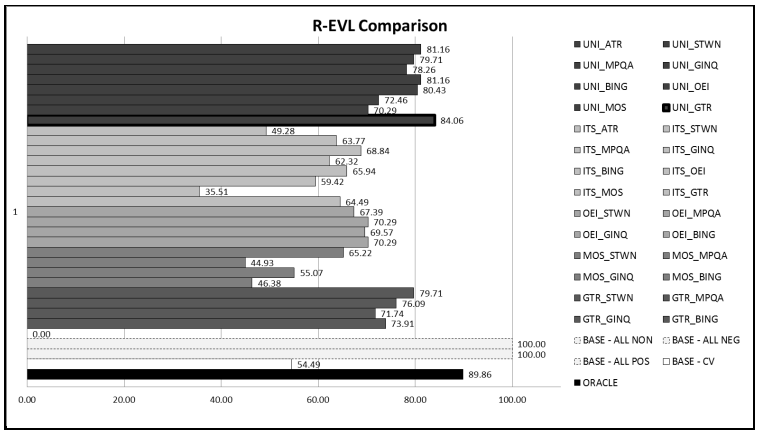


Figure 2.10: R-EVL Comparison of All Lexicons

2.5.6 Adding ITS_GTR to BASE-CV

In this experiment, we added one of the lexicons to the baseline BASE-CV. The purpose of the experiment is to see whether we can gain some improvement on the baseline using this lexicon. We chose ITS_GTR which has the highest precision and accuracy. The expressions from ITS_GTR were added to each lexicon in BASE_CV (2.4.4) and we did 3-fold cross validations as what we did when we measured the BASE-CV. Hence, we got 3 new lexicons: BASEADD1CV12, BASEADD1CV01, and BASEADD1CV02. Figure 2.11 shows the result of our experiment. It can be seen from the graph that adding ITS_GTR to the BASE-CV (BASEADD1-CV) improves the recall quite significantly, from 54.49% to 72.36%, which is beyond ITS_GTR alone (64.49%). However, the precision drops more than the recall increases. Instead of around 70%, we are now back to around 50%.

We tried to analyze this decrease in precision by putting thresholds on the frequencies of the expressions inside the ITS_GTR. The frequencies were gained

from the experiment with frequency scoring function (Section 3.2). We took expressions (for both positive and negative) which have frequencies greater than 100, 75, 50, 25, 10, 5, 1, and 0, producing 8 new different lexicons. Then, we added each of these lexicons to the BASE-CV in the same way as previous experiment producing BASEINC1-CV-THN, where N is the threshold. We expected to see a smooth decrease in the graph by assuming that the expressions with low frequencies cause the performance (precision) to drop significantly. From Figure 2.12, we can see that taking only expressions with frequency higher than 100 (BASEINC1-CV-TH100) does help in increasing the precision compared to BASEADD1-CV, but the drop in precision is already significant. Lowering the threshold also lowers the precision, but the recall increases as the threshold gets lower.

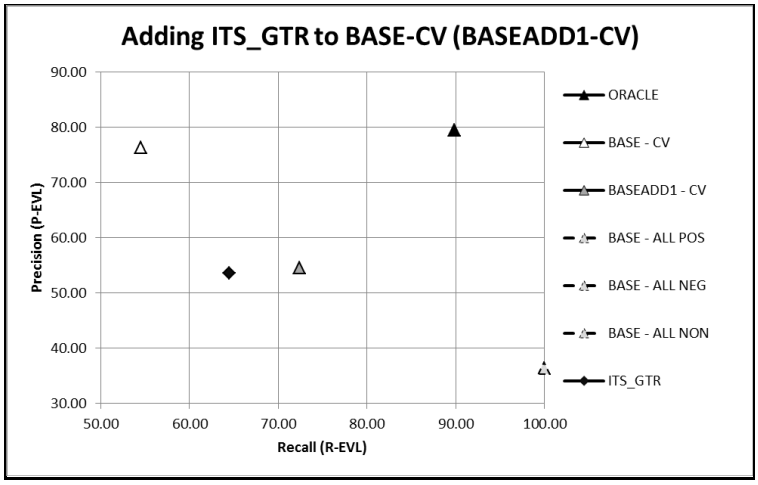


Figure 2.11: Adding ITS_GTR to BASE_CV

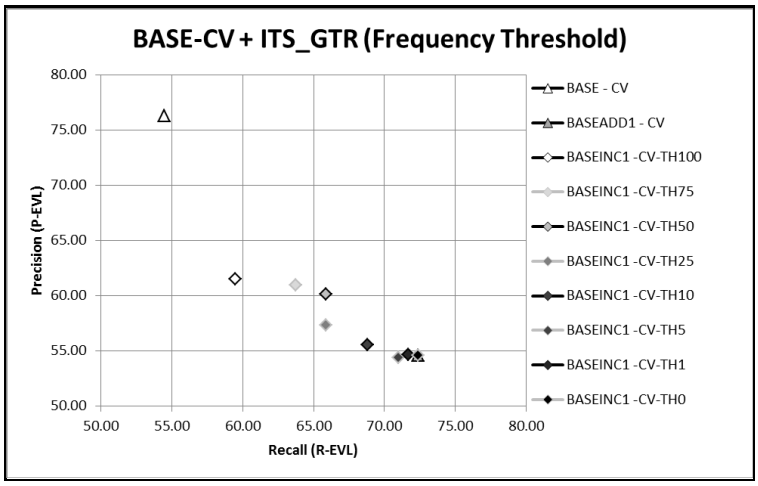


Figure 2.12: Adding the ITS_GTR Thresholded by Frequency

We also tried to sort in descending order the expressions inside ITS_GTR by their frequencies. We took the top N expressions from the lexicon and added them to BASE-CV producing BASEINC2-CV-ADN. The result is shown in Figure 2.13. Using only the top 10 positive and negative expressions causes a significant drop in precision. Adding more words increases the recall but also

lowers the precision. Removing the top 10 positive and negative expressions from lexicon restricted to top 200 expressions (BASEINC2-CV-AD200MN10) improves the precision but lowers the recall even more.

Adding only 1 word at a time from the top 10 (BASEINC2-CV-AD1B1-N) in Figure 2.14 shows that the most problematic expressions are the top 1 expressions, 'cukup' (enough) for the positive one and 'salah' (wrong) for the negative one. This might be related to some of the problems presented in Section 4.1 where these words can be non-evaluative in certain phrases or usages. We show that removing these expressions from the top 10 (BASEINC2-CV-AD10MN1) and top 50 (BASEINC2-CV-AD50MN1) helps to increase the precision while maintaining or increasing the recall. However, for top 200 (BASEINC2-CV-AD200MN1), the precision stays low and only the recall increases.

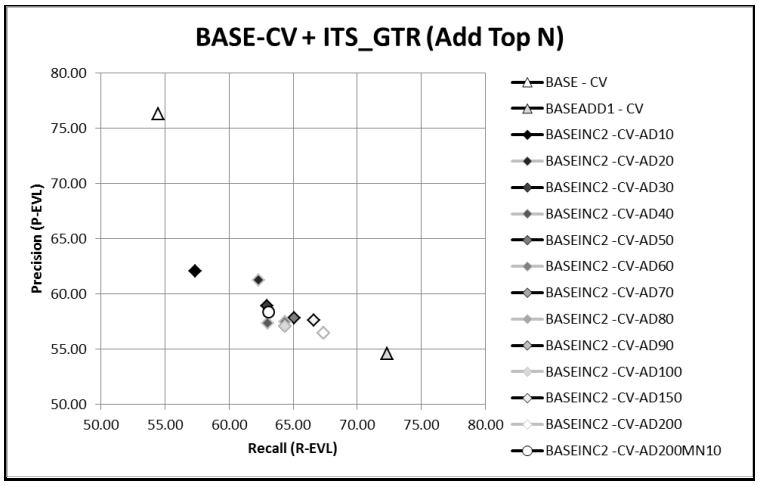


Figure 2.13: Adding the Top N Expressions from ITS_GTR

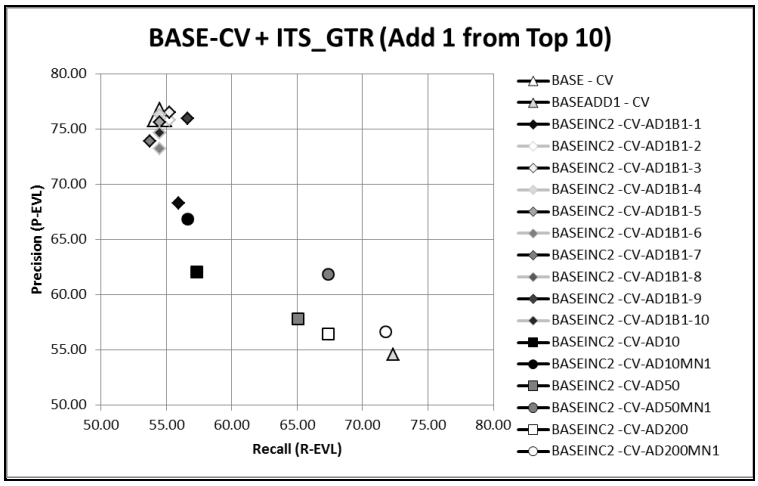


Figure 2.14: Adding One Word from Top 10 of ITS_GTR

2.5.7 Performances and Lexicon Size

We tried to observe the relationship between the size of the lexicon and the performances gained. This is related to our suspicion that the size of the lexicon might

have contribution to the performances of the lexicons. We plotted the performances of basic lexicons and the lexicons from intersection and union operations against their total number of expressions (positive and negative).

We show in Figure 2.15 the relationship between the size of the lexicon (total of positive and negative expressions) and the accuracy gained. The line shows the logarithmic function with least square approximation to the points.We can see from the figure that the accuracy decreases as the number of expressions inside the lexicons increases.

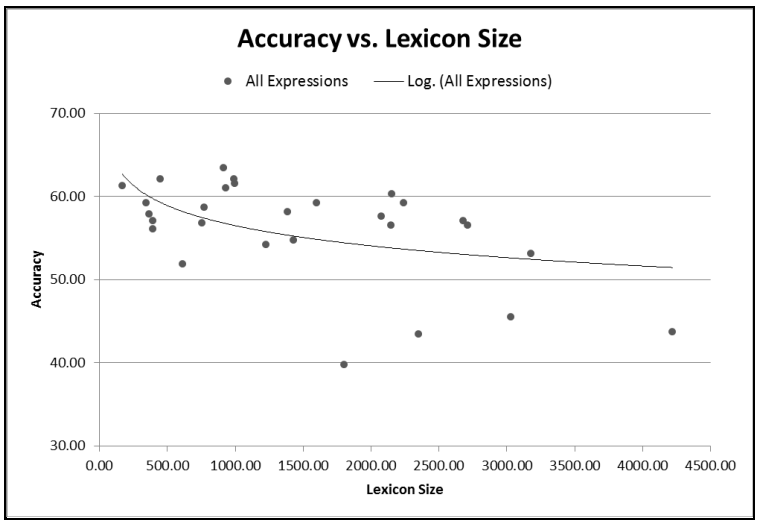


Figure 2.15: Comparison of Lexicon Size and Accuracy

Observing the precision in Figure 2.16, we can see that the precision also decreases as the lexicon size increases. For the recall (Figure 2.17), the behaviour is contrary to the accuracy and precision. As the lexicon size increases the recall also increases.

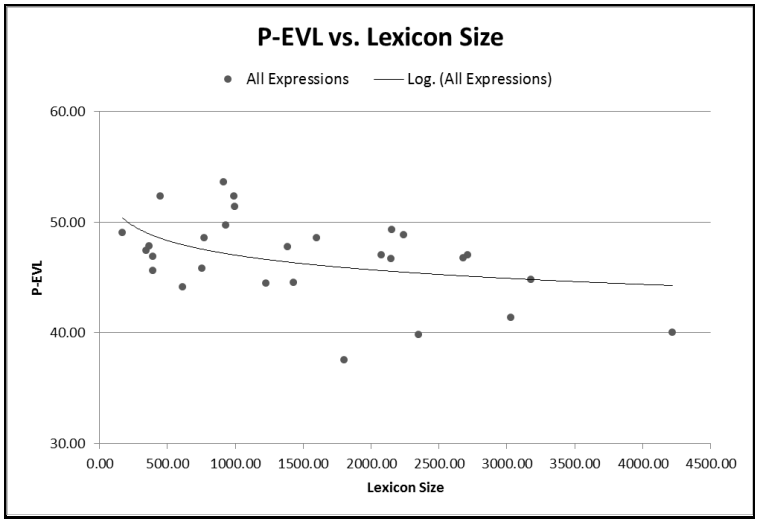


Figure 2.16: Comparison of Lexicon Size and P-EVL

The interesting thing to observe is the increasing and decreasing rate of these measures. The accuracy and precision decrease at slower rate compared to the

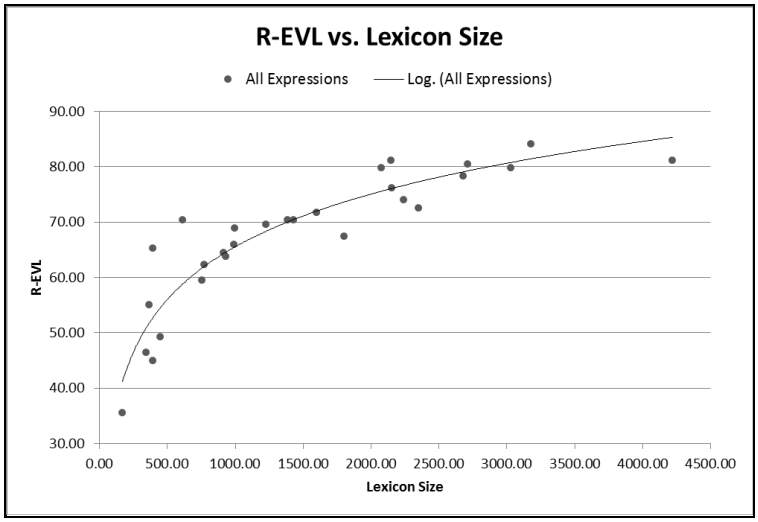


Figure 2.17: Comparison of Lexicon Size and R-EVL

recall, which smoothly increases as the size of the lexicon increases. It means that we can increase the recall quite significantly by adding more expressions while in some sense maintaining the accuracy or precision. Unfortunately, in this case, the accuracy and precision already have low performance when the lexicon size is small.

In Figure 2.18 we can see the cause of this behaviour. The flat line shows the total number of evaluative sentences. Because the number of sentences predicted as evaluative increases as the size of the lexicon increases and given the fixed number of evaluative sentences, the precision gets lower. On the other hand, the recall gets higher as the lexicon size increases, and given the small number of evaluative sentences, the recall measured in percentage grows fast.

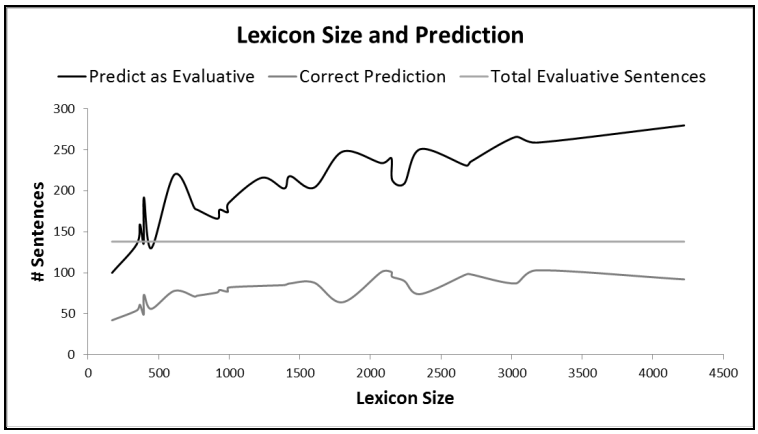


Figure 2.18: Comparison of Lexicon Size and the Correct Prediction

3. Weighting the Lexicons

The purpose of weighting is to measure the importance of the expressions. Despite their subjectivity, some expressions might be weaker or stronger indicators of sentence polarity. We tried to measure this importance by giving a weight to each expression based on several weighting methods.

3.1 Selecting Lexicon for Weighting

We chose one of the lexicons from the translations, intersections, and unions operations to be worked on. The selection process started by looking at all measures (accuracy, precision, recall, F1) and we took lexicons that scored in the top five more than four times. This gave us 8 lexicons as shown in Table 3.1.

Table 3.1: Selecting Lexicons for Weighting

Measure	GTR_ MPQA	GTR_ STWN	ITS_ GTR	ITS_ MPQA	ITS_ ALL	UNI_ GTR	UNI_ BING	UNI_ GINQ
Accuracy	60.26%	57.63%	63.42%	61.58%	62.11%	53.16%	56.58%	56.58%
P-EVL	49.3%	47.01%	53.61%	51.35%	52.31%	44.79%	47.03%	46.67%
R-EVL	76.09%	79.71%	64.49%	68.84%	49.28%	84.06%	80.43%	81.16%
F1-EVL	59.83%	59.14%	58.55%	58.82%	50.75%	58.44%	59.36%	59.26%
P-NON	80.24%	80.82%	77.1%	77.95%	72%	81.82%	81.25%	81.43%
R-NON	55.37%	48.76%	68.18%	62.81%	74.38%	40.91%	48.35%	47.11%
F-NON	65.53%	60.82%	72.37%	69.57%	73.17%	54.55%	60.62%	59.69%
P-POS	54.32%	49.22%	61.74%	55.56%	57.14%	45.75%	53.14%	51.34%
R-POS	70.4%	76%	56.8%	60%	41.6%	77.6%	74.4%	76.8%
F-POS	61.32%	59.75%	59.17%	57.69%	48.15%	57.57%	62%	61.54%
P-NEG	13.73%	14.63%	9.8%	14%	10.26%	12.77%	8.2%	9.43%
R-NEG	53.85%	46.15%	38.46%	53.85%	30.77%	46.15%	38.46%	38.46%
F-NEG	21.88%	22.22%	15.63%	22.22%	15.38%	20%	13.51%	15.15%

The evaluation results of the lexicons coming from Google Translate (GTR) and union operations (UNI) show that they have imbalanced results on their precisions and recalls for evaluative sentences and non-evaluative sentences. For example, the P-EVL and R-EVL or P-NON and R-NON of GTR_MPQA in Table 3.1 show that their performances bias towards the recall (in EVL) or precision (in NON) with significant difference between the precision and recall values. While this behaviour also exists on the results from intersection operations (ITS), the difference is more modest. Hence, we chose to work with the lexicons from the intersection (ITS) operations. We decided to use ITS_GTR lexicon for the weighting since it is better than ITS_ALL in most cases, although it has comparable performance with the ITS_MPQA.

3.2 The Weighting Functions

We defined two weighting functions based on the frequency and relative frequency (iterative weighting) of the expressions in the training data. The training data is a collection of 14,998 unannotated sentences extracted from the reviews (Section 2.2.1).

3.2.1 Frequency Weighting

The first weighting function is based on the frequency of the expressions in the training data. The basic idea is that in reviews, evaluative expressions should be frequent. If a term from the generic lexicon appears rarely in the texts, it is probably not an evaluative expression in the given domain. The weights of positive expression p and negative expressions n are given based on the following formula:

$$weight_{pos}(\mathrm{p}) = freq_{all}(\mathrm{p}) \tag{3.1}$$
$$weight_{neg}(\mathrm{n}) = freq_{all}(\mathrm{n}) \tag{3.2}$$

The $weight_{pos}(\mathrm{p})$ is the weight of positive expression and $weight_{neg}(\mathrm{n})$ is the weight of negative expression. The $freq_{all}$ is the total number of occurrences of the expression in the training data. We assumed that the expressions are always used in evaluative manner and possess positive or negative sense. This assumption is a very rough one as evaluative words might be evaluative or not evaluative depending on the contexts where they are used in.

3.2.2 Iterative Weighting

In iterative weighting, the weights of the expressions are calculated as the proportion of occurrence of the positive or negative expressions in their respective positive or negative sentences in the training data. Since the training data has no positive and negative annotation, we used the simple prediction method (Section 2.4.1) to predict the sentiments of the sentences. The initial weight of each lexicon entry is 1.0. At the end of the annotation process, we calculated the weights of positive expression p and negative expressions n as:

$$weight_{pos}(\mathrm{p}) = freq_{pos}(\mathrm{p})/freq_{all}(\mathrm{p}) \tag{3.3}$$
$$weight_{neg}(\mathrm{n}) = freq_{neg}(\mathrm{n})/freq_{all}(\mathrm{n}) \tag{3.4}$$

The $freq_{pos}(\mathrm{p})$ is the frequency of the positive expression in sentences marked as positive and $freq_{neg}(\mathrm{n})$ is the frequency of the negative expression in sentences marked as negative. We repeated the annotation (with the new weights) and reweighted the expressions until there were no changes in the weights (convergence).

3.3 Evaluating the Lexicons

We evaluated the resulting lexicons from the two weighting functions above using the same test data and the simple prediction method as used in Chapter 2.

3.3.1 Frequency Weighting Results

We compared the results of the predictions using the lexicon with frequency weighting (ITS_GTR_FREQ) and the source lexicon with default weight 1.0 (ITS_GTR) in Figure 3.1. From the figure, the accuracy and precision of the new lexicon are lower than the original lexicon, although not significantly lower. However, the recall of the lexicon with frequency weighting increases by around 9% absolute. The lexicon with frequency weighting seems to break some ties in sentiment prediction and it is able to detect more evaluative sentences. However, this is not followed by the increase in the precision and

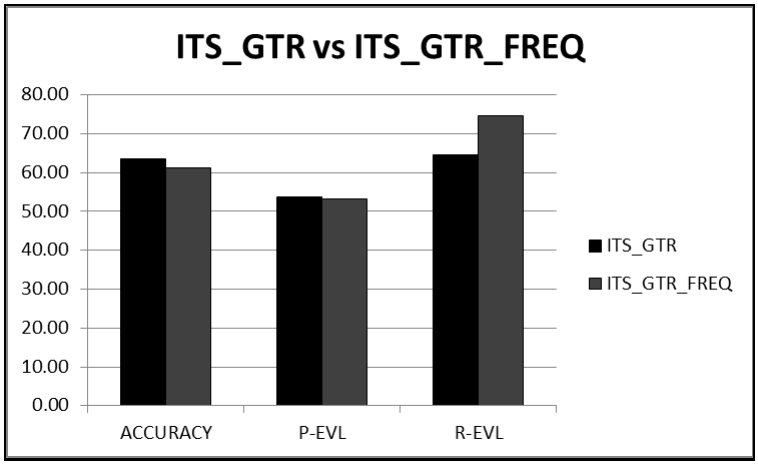


Figure 3.1: Comparison of ITS_GTR and frequency weighting

accuracy, which means that the given weights also cause more non-evaluative sentences to be predicted as evaluative.

We also tried to look at the results of the prediction by using only the expressions with weights (frequencies) greater than a certain threshold. Figure 3.2 shows the results of the experiment. It can be seen from the figure that using a higher threshold causes an increase in the accuracy and precision values, but in a small margin. On the other hand, the recall seems to drop much faster than the accuracy and precision grow.

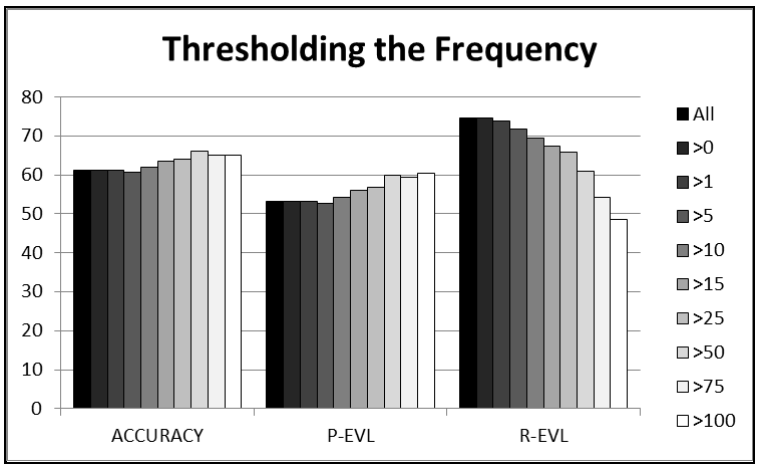


Figure 3.2: Comparison using Different Threshold

3.3.2 Iterative Weighting Results

Figure 3.3 shows the prediction results of lexicon at each iteration. The iterative weighting converges at iteration 5. We can see from the results that there are no changes in the prediction performances from each iteration. Except for a small improvement in prediction from iteration 1 to 2.

We tried to look at the number of expressions that change their weights at each iteration and also the changes in the sentiment annotation of the sentences in training data. Table 3.2 shows these numbers. There is a large change in the number of expression weights and sentiment of the sentences in the first iteration. This probably

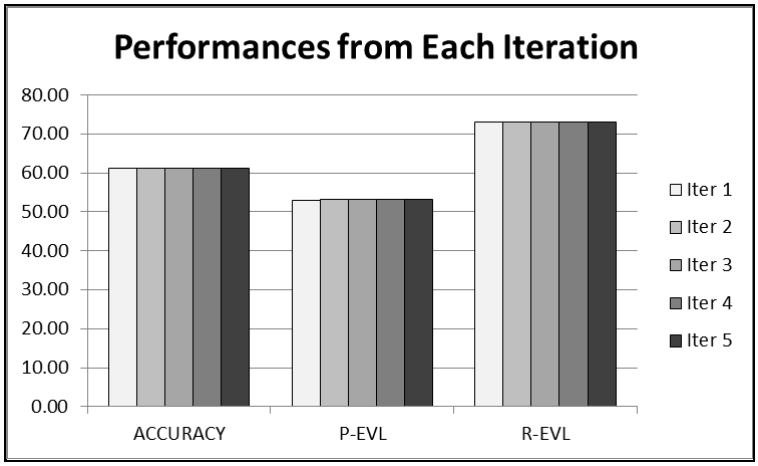


Figure 3.3: Comparison of Performances from Each Iteration

can be seen from the small change in precision from iteration 1 to 2. The changes in the later iterations are smaller.

From iteration 1 to 2, all of the changed expressions have their weights increased. This could explain the little change in performances despite many expressions changed their weights from iteration 1 to 2.

Table 3.2: Number of Changes in Sentiment Annotations and Weights

Changes of	Iterations			
	1-2	**2-3**	**3-4**	**4-5**
# Sentiment of Sentences	984	36	11	2
# Weights in Positive Lexicon	157	28	6	3
# Weights in Negative Lexicon	83	28	6	1

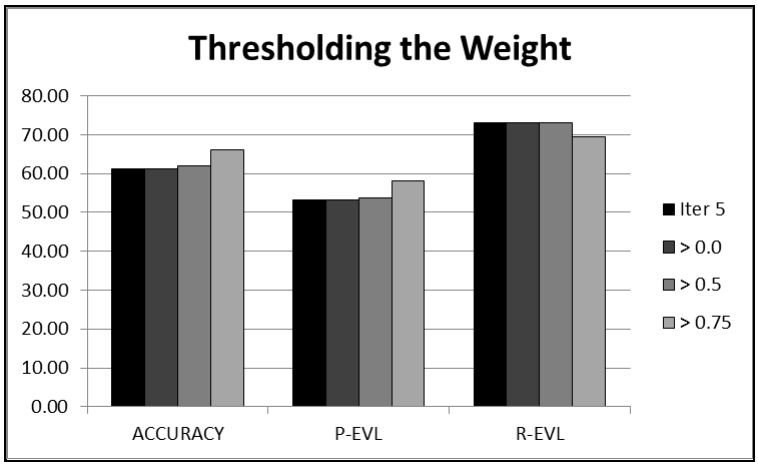


Figure 3.4: Performances of ITS_GTR_ITER using Different Threshold

From iteration 2 to 3, the tendency is for positive expressions to increase their weights and for negative expressions to decrease theirs. We tried to observe the prediction results to find out why the performances are still the same despite this behaviour. We found that there are only 43 out of 380 sentences that contain both positive and negative expressions, which means that there are only around 43 sentences that might

change their polarities. By looking into the prediction results of these sentences, our current conclusion is that the changes on the weights are mostly very small and do not give enough impact to the change in prediction. For the later iterations, the number of expressions that change their weights are too small and hence, do not give significant changes to the results.

We tried to look at the behaviour of using only the expressions that have weights above a certain threshold. The results are shown in Figure 3.4. We observe an increase in accuracy and precision as the threshold increases, especially from threshold 0.5 to 0.75. But, unfortunately, this increase in accuracy and precision is accompanied by a comparable decrease in recall.

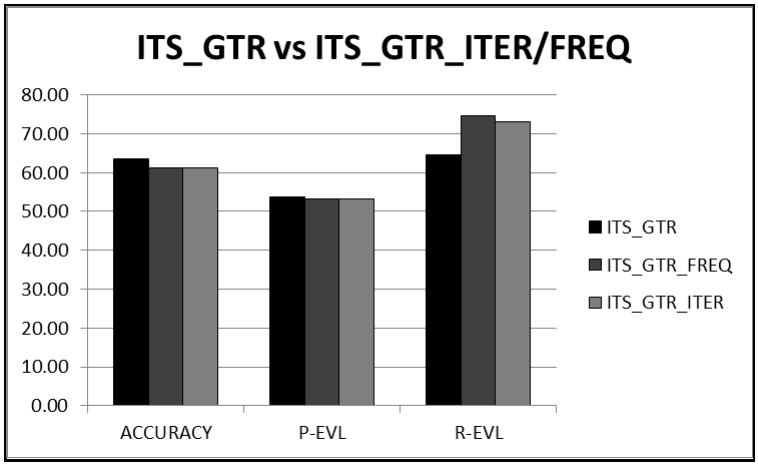


Figure 3.5: Comparison of ITS_GTR, frequency, and iterative weighting

The comparison of the lexicon from iterative weighting (ITS_GTR_ITER) with the lexicon using default weighting shows that the iterative weighting leads to a lexicon with a higher recall and a lower accuracy and precision (Figure 3.5). The comparison with the lexicon from frequency weighting shows that iterative weighting has the same performances in accuracy and precision, but it is lower in recall.

3.3.3 Iterative Weighting with Thresholding in Prediction

In the following experiments, we slightly modified the simple prediction to incorporate a threshold, so that a sentence will be marked as evaluative (positive or negative) if the total weight of positive or negative expressions is higher than the given threshold. The reason is to get a stricter prediction with the expectation that the resulting annotations on the unannotated sentences are closer to the correct or true annotations, and in turn produce better weights for the expressions.

Given that s is the sentence to be predicted that contains a list of positive expressions P and a list of negative expressions N, and t is the given threshold, the prediction function becomes:

$$polarity(\mathrm{s}) = \begin{cases} positive & \sum_{p \in P} weight_{pos}(\mathrm{p}) > \sum_{n \in N} weight_{neg}(\mathrm{n}) \text{ and } \sum_{p \in P} weight_{pos}(\mathrm{p}) > t \\ negative & \sum_{p \in P} weight_{pos}(\mathrm{p}) < \sum_{n \in N} weight_{neg}(\mathrm{n}) \text{ and } \sum_{n \in N} weight_{neg}(\mathrm{n}) > t \\ neutral & \sum_{p \in P} weight_{pos}(\mathrm{p}) = \sum_{n \in N} weight_{neg}(\mathrm{n}) \end{cases}$$

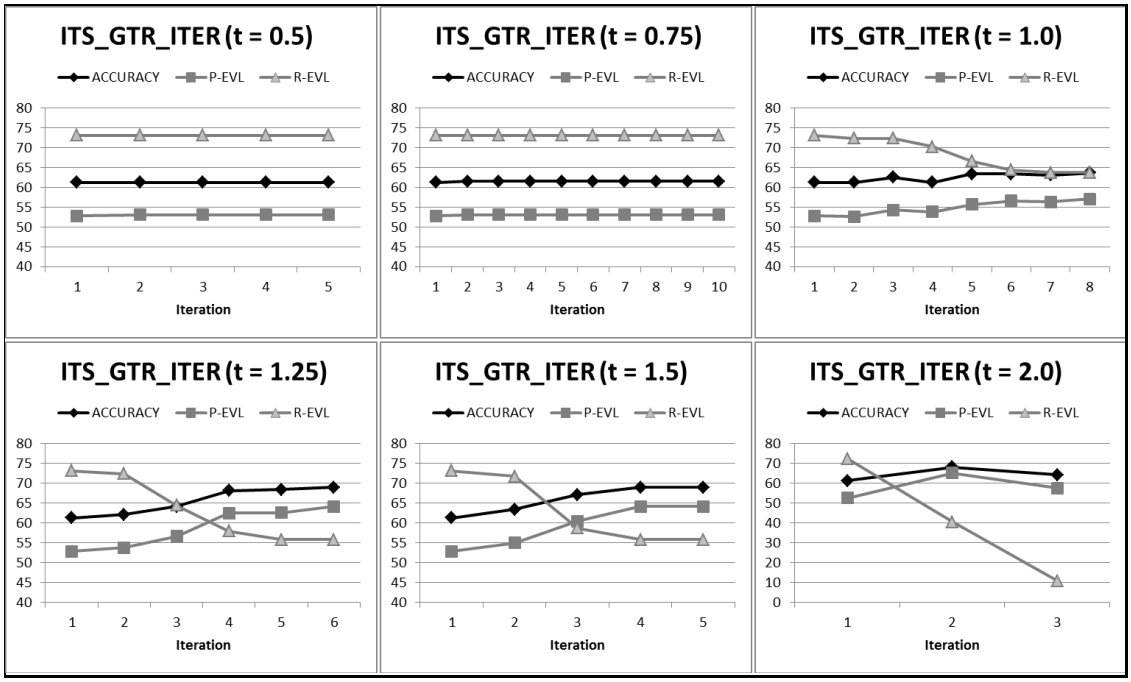


Figure 3.6: Comparison of from Each Iteration using Different t

This modification is only used in the weighting process for training data, while for the prediction for evaluation purpose, we still use the basic simple prediction method.

We show in 3.6 the movement of accuracy, precision, and recall for each iteration using different t. Using $t = 0.5$ and $t = 0.75$, the results are steady and do not change for all iterations. This behaviour is the same as the iterative weighting using the unmodified simple prediction method. For other values of t, the accuracy and precision get higher and the recall get lower throughout the iterations. The recall drastically drops at $t = 2.0$ (note the different range of the y-axis).

We looked at the resulting lexicons to see why this behaviour occurred. It turned out that as the t increases the number of expressions that get the weight greater than 0.0 decreases. In the extreme case as $t = 2.0$, this causes the resulting lexicons to only have very few (less than 10) expressions that have non-zero weight.

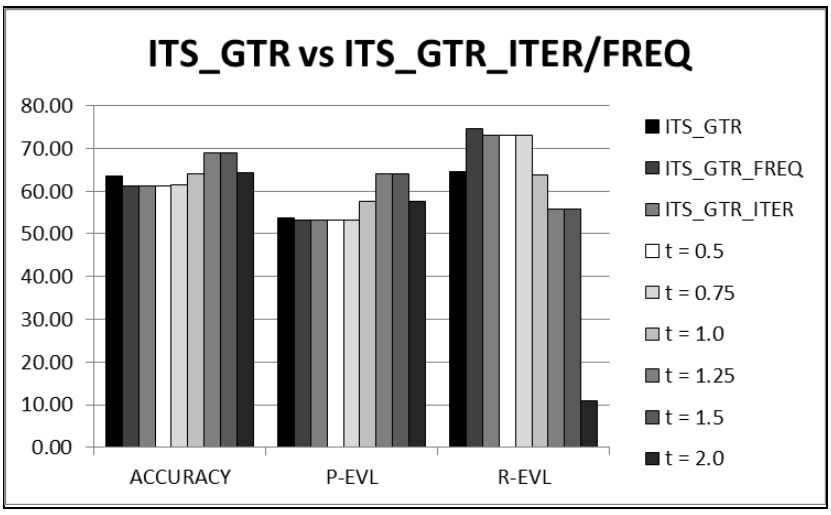


Figure 3.7: Comparison of ITS_GTR, frequency, and iterative weighting

Comparing all of the resulting lexicons from the weighting up to this point (Figure 3.7), we can see that putting the threshold $t = 1.25$ and $t = 1.5$ seem to successfully increase the accuracy and precision around 10%. But, the problem with the decrease in the recall still exists. The recall decreases more than the accuracy and precision increase.

4. Predicting using Machine Learning

In this chapter, we will explain our experiments that use machine learning method to do the prediction of the sentiment. In the previous chapters, the prediction was performed using a simple comparison of the number of positive expressions and negative expressions inside the sentence. In the case of weighting, the comparison used the total weight of positive and negative expressions. The purpose of changing the prediction function is to handle some weaknesses of the previous prediction method. Specifically, the previous method does not consider information like part-of-speech information and depends solely on the lexicon. We wanted to see if adding some of this information as features in a machine learning setup can bring an improvement.

4.1 Defining Features

In order to realize what kind of information the simple prediction lacked, we took the prediction results of sentences in CV0 that was tagged using BASECV12 and BASEADD1CV12 (2.5.6). The BASECV12 contains the list of expressions from data in CV1 and CV2. BASEADD1CV12 contains the list of expressions from BASECV12 added with expressions from ITS_GTR. We compared the tagging results using these two lexicons to see what causes the lower performance in BASEADD1CV12 and also to see what causes the errors in predictions. In general, we found several types of information that the simple prediction method of comparing numbers of positive and negative expressions cannot derive.

- Sentence Structure
  A sentence with positive or negative expressions might not always be an evaluative sentence. The expressions can be non-evaluative or depending on the sentence structure, the overall sentiment might be neutral. From the observations, we found several things that might cause the evaluative expressions to have no effect on the overall sentiment.

  The first case is when the sentence is in a hypothetical form, as in the example of 'sebuah keputusan yang salah akan membuat jiwa seluruh batalyon melayang percuma' (one wrong decision will cause the death of all batallions). In this sentence the word 'salah' (wrong) is identified as negative expression. However, this is only a hypothetical situation where the speaker expresses the opinion of what will happen, but not to evaluate the decision itself.

  Another structure that might affect the sentiment of the sentence is when it contrasts the positive and negative expressions as in the examples below:

  'menyuguhkan fitur yang berbeda, walau dengan model yang sama' (it comes with different features, though with the same design/model)

  'walau dengan model yang sama, menyuguhkan fitur yang berbeda' (though it has the same design/model, it comes with different features)

  In this context, the word 'berbeda' (different) is positive and 'sama' (same) is negative. Changing the parts of the sentence that are separated by a comma (one with 'though' and one without 'though') and depending on where the positive

and negative expressions are, the sentiment of the sentences can be different. The first sentence seems to be neutral and the second one seems to be more positive.

Questions are also mostly neutral, e.g., 'butuh ponsel yang murah tapi meriah?' (need a cheap and fancy phone?). The occurrence of evaluative expressions 'murah' (cheap) and 'meriah' (fancy) have no effect on the final sentiment of the sentence.

- Lexical Information
  Some other information that can be useful is related to the word itself. The first such piece of information is the part-of-speech of the word. Some evaluative expressions might have a different meaning depending on what part-of-speech they take in a sentence. For example, the word 'menarik' can have meanings of 'pull' (verb), which can be considered as having no sentiment, and 'interesting' (adjective) which has positive sentiment.

  The other thing is that the evaluative expressions are sometimes used in a non-base form, e.g., 'indahnya' (how beautiful), which has the base form of 'indah' (beautiful). A simple word matching without lemmatization or stemming might not be able to capture the evaluative expression.

  Words that are part of larger phrases are also tricky and might cause an inappropriate detection of evaluative expressions, e.g., 'kurang lebih' (more or less), which contains the word 'kurang' (not enough) and 'lebih' (more/better). Predictions with simple word matching that we used in previous experiments are not able to capture this phrasal information.

- Target
  Information about the target of the discussion or target of the evaluation in an evaluative sentence is also important. Some sentences contain evaluative expressions that are not targeted at the main target of the discussion, e.g., 'selain bisa untuk berbelanja, website.com ... dengan foto-foto bayi anda yang lucu' (in addition to shopping, website.com ... with photos of your cute babies), where the target of the discussion is 'website.com' but contains a positive expression 'cute' for another target, the object 'baby'.

We defined a small set of 12 binary features which consists of 10 non-lexicon related features (NonLexFeats) and 2 lexicon related features (LexFeats) based on these observations. The two lexicon related features tell whether at least one expression from the positive or the negative lexicon was seen in the sentence. The non-lexicon related features capture the information as explained above. We did not include any features concerning the target of the discussion, as the detection of the target from the sentences is very difficult. The features defined are listed in Table 4.1.

We differentiated between 'Adjective Word' and 'Adjective Lemmas', since in Indonesian, an adjective can be changed into another part-of-speech using appropriate prefix/suffix as in 'indah' (beautifull) and 'ke*indah*an' (beauty), or from other part-of-speech to adjective as in 'ibu' (mother) and 'ke*ibu*an' (motherly). As for 'Question Word' and 'Question Lemma', we differentiated them since the question word can be used in its base or non-base form, as in 'apa' (what) and '*apa*kah' (what).

Table 4.1: Features for Sentiment Prediction

Name	Type	Set to True if
Hypothetical	NonLexFeats	Any of the words 'jika' (if), 'akan' (will), 'kalau' (if) appears in the sentence
Question	NonLexFeats	'?' (question mark) appears in the sentence
Contrast 1	NonLexFeats	Any of 'walaupun', 'meskipun', 'walau', 'meski' (though/although) is the first word of the sentence
Contrast 2	NonLexFeats	Any of 'walaupun', 'meskipun', 'walau', 'meski' (though/although) appears anywhere except the first/last word
Negative List	NonLexFeats	Any of the phrases 'cukup sampai disitu' (only until that point), 'kurang lebih' (more or less), 'salah satu' (one of the) appears in the sentence
Negation List	NonLexFeats	Any of the words 'tidak' (not), 'tak' (not), 'tanpa' (without), 'belum' (not yet), 'kurang' (less), 'bukan' (is not) appears in the sentence
Adjective Word	NonLexFeats	Any adjective (surface) words appears in the sentence
Adjective Lemma	NonLexFeats	Any adjective lemmas appears in the sentence
Question Word	NonLexFeats	Any question (surface) words, e.g. 'apakah' (what), 'bagaimanakah' (how), appears in the sentence
Question Lemma	NonLexFeats	Any question lemmas, e.g. 'apa' (what), 'bagaimana' (how), appears in the sentence
PosLex	LexFeats	At least one of the positive expressions from the lexicon appears in the sentence
NegLex	LexFeats	At least one of the negative expressions from the lexicon appears in the sentence

4.2 Implementation Details

For the experiments, we used *scikit-learn*[1], a machine learning library for Python. We chose SVM as the machine learning method and used the default function *svm.SVC()* provided by the library. The kernel used by this function is an RBF kernel, and we just used the function with its default parameters as shown in Figure 4.1.

We only changed the parameter of *class_weight* to *auto* to handle the different size (imbalance) of positive, negative, and neutral sentences in training data. The weight given to each class is inversely proportional to the class frequencies in the training data[2].

The information of part-of-speech tags and lemmas is provided by MorphInd[3], a

[1] `http://scikit-learn.org`

[2] `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC`

[3] `http://ufal.mff.cuni.cz/~larasati/morphind/index.html`

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, degree=3,
gamma=0.0, kernel='rbf', max_iter=-1, probability=False,
shrinking=True,  tol=0.001, verbose=False)
```

Figure 4.1: Default Parameters of SVC Function in Scikit-learn

robust finite state morphology tool for Indonesian [17]. We extracted the word tags, lemmas, and lemma tags to be added to our training sentences, and used them in feature extraction process.

4.3 Evaluation Results

The evaluation was performed using 3-fold cross validation with the same division of the data as in previous chapters. The results shown here are the average value across the three runs. We used the same naming convention for the lexicon with additional '_ML' marking at the end of each lexicon name. As the features were defined based on our observation of CV0, there is a risk of the prediction results of CV0 will be much higher than on the other two folds. From our observation, the prediction results on the three folds are not significantly different, and although the features are defined based on some findings in CV0, they do not seem to leverage the performance on CV0 to have the highest performance. Therefore, we included CV0 in the average rather than reducing on the test set by one third.

4.3.1 Comparing Different Categories of Features

We first compared the different categories of features for each lexicon. We did comparisons on two categories of features, AllFeats that used all the features available, and LexFeats that used only two features (PosLex, NegLex). Figure 4.2 shows the comparison of accuracy when using these two types of features for each lexicon.

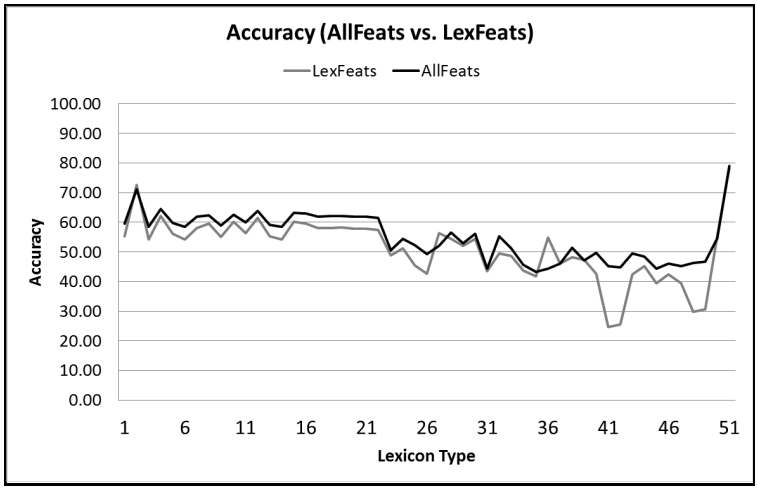


Figure 4.2: Comparison of Accuracy using Different Features Category

The x-axis represents the type of lexicon. Looking at the graph, we can see that almost all the lexicons have more or less the same performance for AllFeats and LexFeats, although using all the features seems to increase the accuracy by a small margin. The significant differences can be seen on lexicon type 41, 42, 48, and 49, which correspond to the lexicons OEI_STWN, UNI_ATR, UNI_STWN, and UNI_OEI. It seems that the

combination of the online dictionary translation with the source lexicon of SentiWordNet gives a lower performance when we only depend on the lexicons as the features. Our current hypothesis is that these lexicons contain a large number of expressions, so that they tend to predict more sentences as evaluative. This behaviour causes the decrease of performance in non-evaluative sentences prediction.

The comparison of precision of predicting evaluative sentences (P-EVL) in Figure 4.3 shows the same behaviour. However, in this case, all lexicons have almost the same precisions without any significant differences. Using all the features seems to only help increase the precision values by a very small margin only.

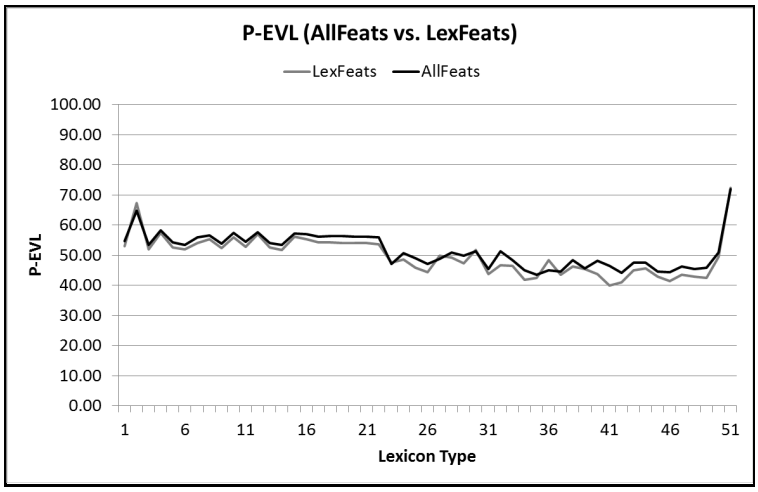


Figure 4.3: Comparison of P-EVL using Different Features Category

The comparison of the recall (R-EVL) values in Figure 4.4 shows that using only LexFeats actually increases the recall by small percentage. The exceptions are in the points 27, 31, 34, 35, and 36 which correspond to ITS_ATR, ITS_MOS, MOS_BING, MOS_GINQ, and MOS_MPQA. This behaviour might be explained by the small number of expressions on each lexicon.

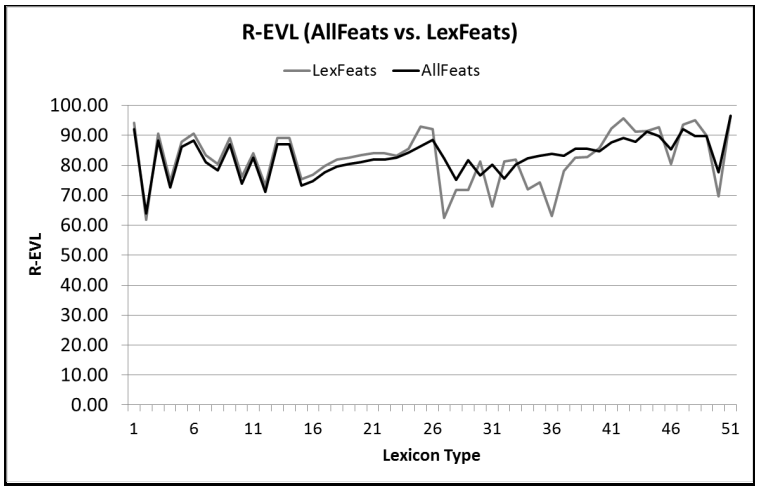


Figure 4.4: Comparison of R-EVL using Different Features Category

The low accuracies near point 41 in Figure 4.2 do not seem to be reflected in the precisions of predicting evaluative sentences (P-EVL). We show in Figure 4.5 that these low accuracies are mostly affected by the low recalls in predicting non-evaluative

sentences. Since non-evaluative sentences constitute most of the test data, this low recall (low number of correct prediction) of non-evaluative sentences greatly affects the overall accuracy.

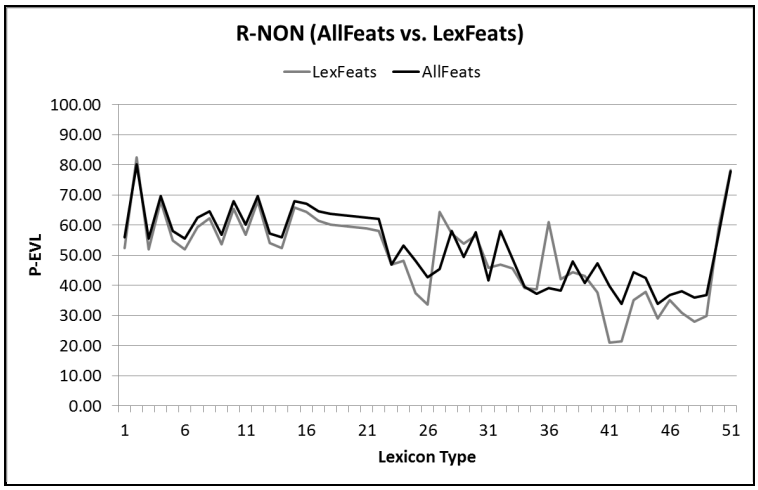


Figure 4.5: Comparison of R-NON using Different Features Category

From Figure 4.6, by averaging the accuracy, P-EVL, and R-EVL of all lexicons using different category of features, we can see that using all the features does help a little in increasing the performance, although, very little. We also show the accuracy, P-EVL, and R-EVL of using only NonLexFeats category which consists of only one experiment, since we do not use any lexicons. Fortunately, the lexicons are at least somewhat useful - not using them reduces performance in all measures.

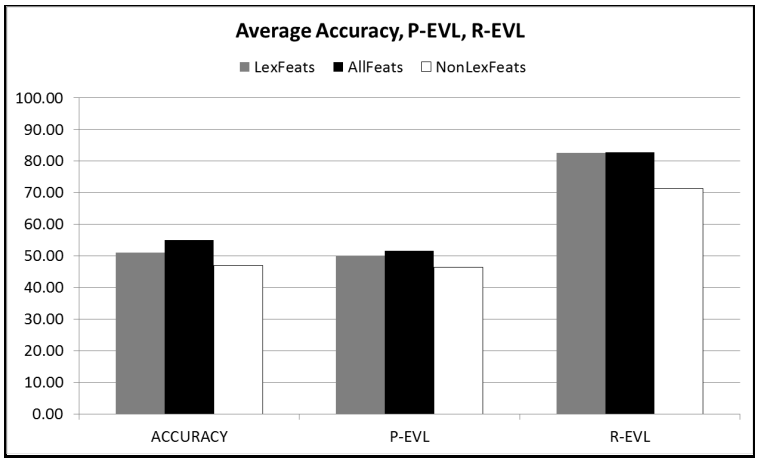


Figure 4.6: Comparison of Average Performances using Different Features Category

We can also see from the graph that the difference in performances between LexFeats and AllFeats, and NonLexFeats are quite small. This might be because we only use two features to represent the lexicon. The simplification of using only two features might not help much in improving the performances.

4.3.2 Comparing the Lexicons

We show in Figure 4.7, Figure 4.8, and Figure 4.9 the comparison of performances of the lexicons using AllFeats. In terms of accuracy, we can see that the baseline BASE_CV

and ORACLE are still better compared to all the lexicons. The non-baseline lexicon with the highest accuracy is ITS_BING.

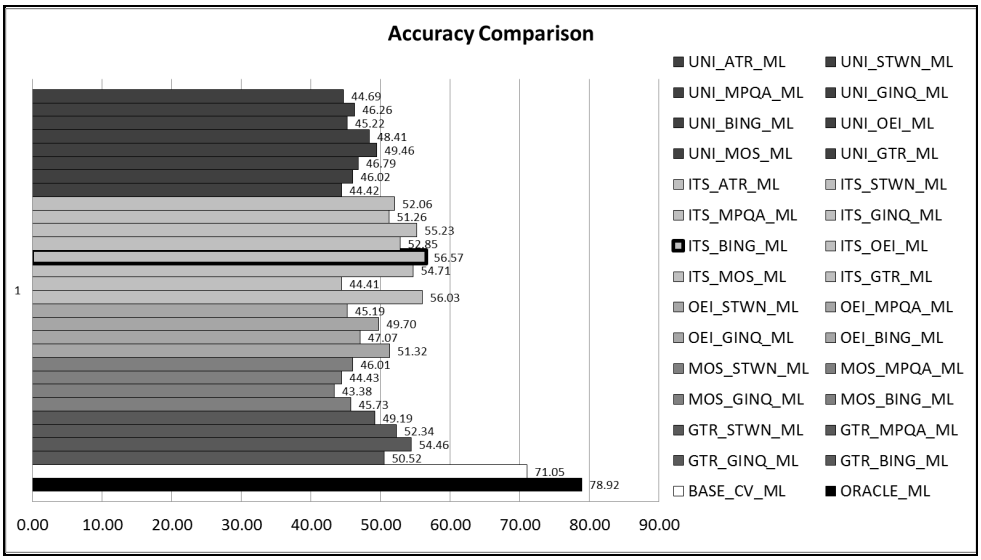


Figure 4.7: Comparison of Accuracy across different lexicons

Looking at the precision (P-EVL), the highest performance is achieved by ITS_GTR, which is still lower than the baselines. However, for the recall (R-EVL), all lexicons manage to surpass the BASE_CV baseline, though they are still lower than the ORACLE. The highest value of recall (R-EVL) is achieved by UNI_MPQA.

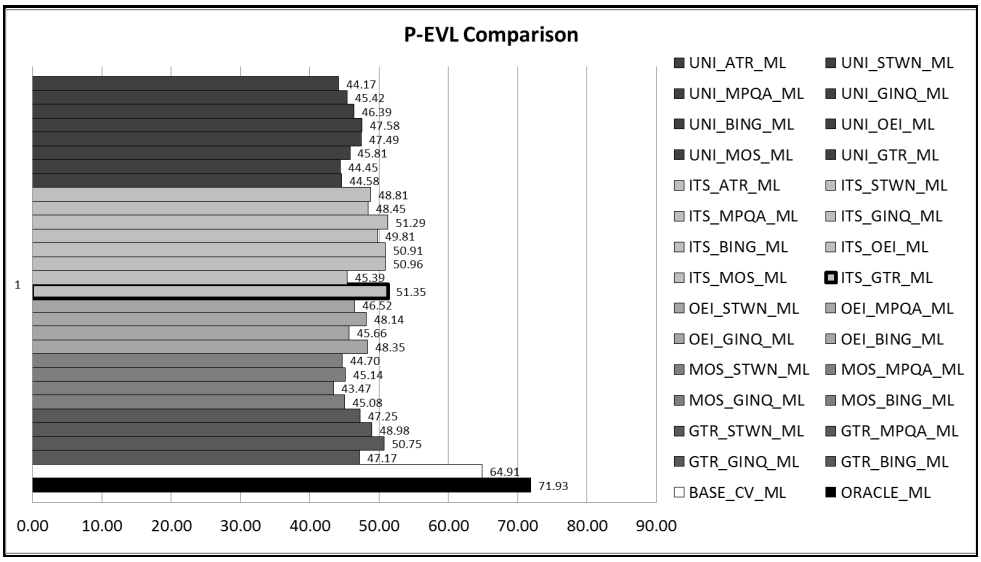


Figure 4.8: Comparison of P-EVL across different lexicons

4.3.3 Precision-Recall Graph

We plot the precision and recall (P-EVL and R-EVL) of all lexicons to see their position in precision-recall graph (Figure 4.10). It is interesting to see that now the precision and recall of the lexicons are closer to each other compared to the precision and recall of the lexicons when using simple prediction method. The behaviour of the lexicons from

intersection and union operations are still the same, with lexicons from intersection tend to have higher precisions and lower recalls, and lexicons from union operations tend to have higher recalls and lower precisions. However, from the figure, we can see that although the recall of each lexicon is high, their precisions are still much lower than of the baselines.

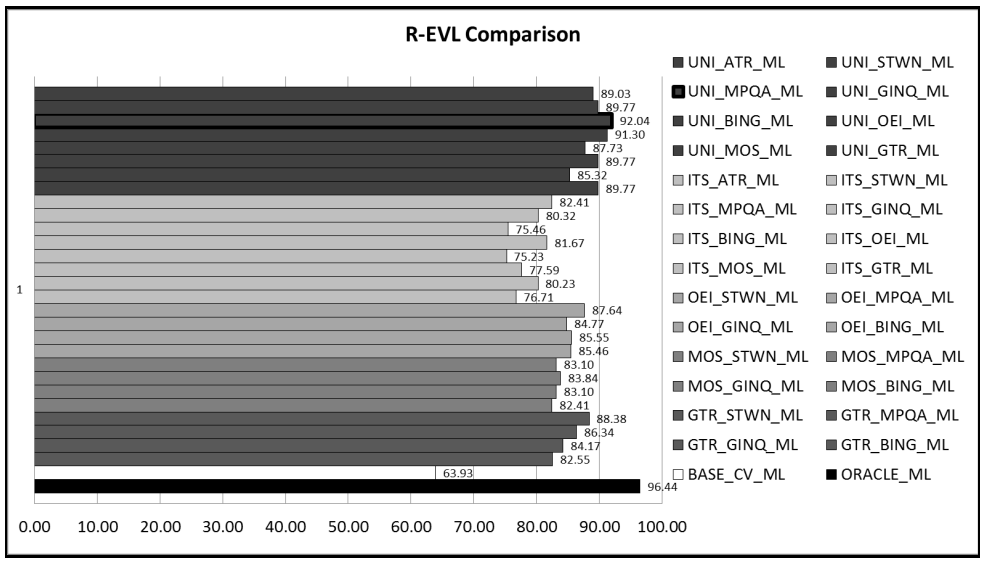


Figure 4.9: Comparison of R-EVL across different lexicons

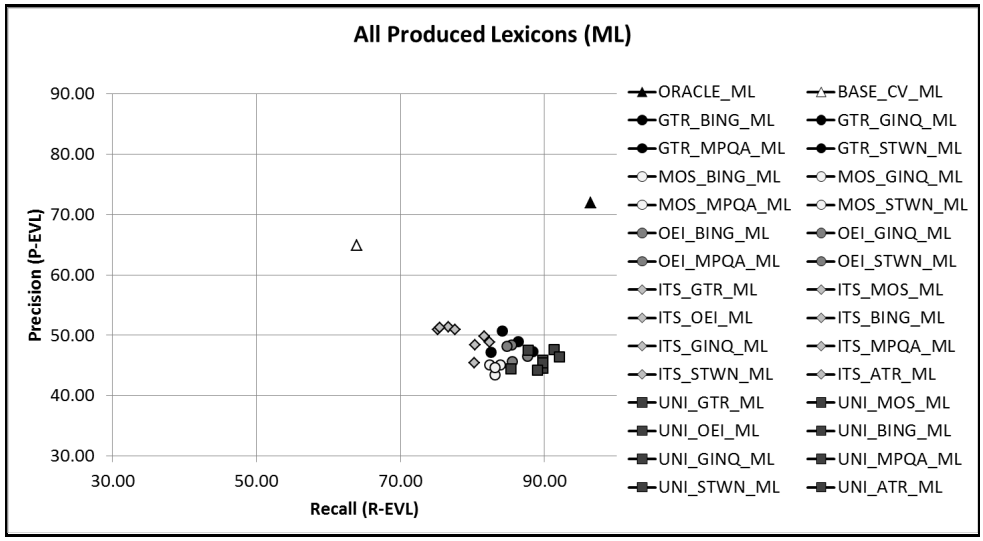


Figure 4.10: Precision-Recall Graph of Lexicons using Machine Learning Prediction

4.3.4 Size of Lexicons

Figure 4.11, 4.12, and 4.13 show the relationship between the size of the positive and negative lexicons with the accuracy, precision, and recall. The lines show the least-square fit of the logarithmic function. We plotted the same lexicons (basic lexicons, intersection, union) as in Section 2.5.7. We can see that the accuracy and precision are

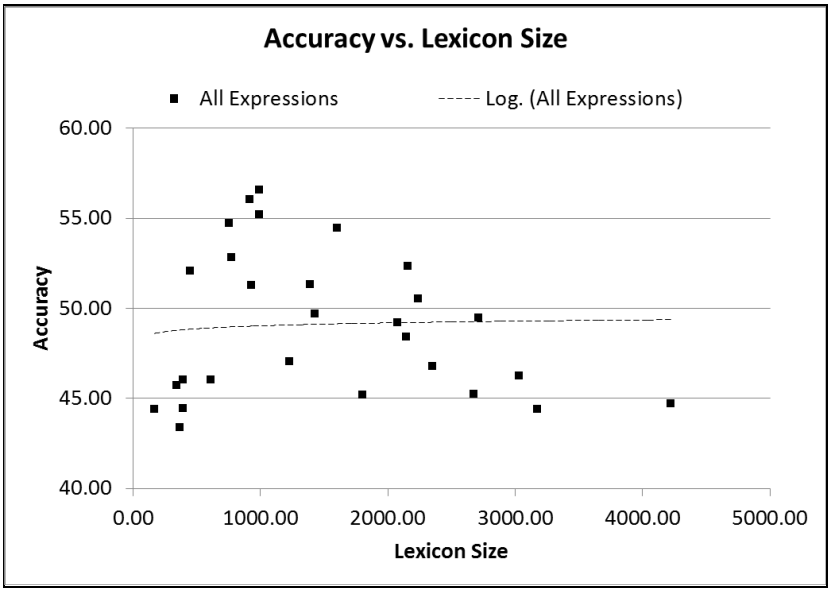


Figure 4.11: Comparison of Lexicon Size and Accuracy

quite flat since the points are scattered, but there is a tendency for the accuracy and precision to be lower at higher lexicon size.

The behaviour of the recall is contrary to the accuracy and precision and it gets higher as the size of the lexicons increase, which is the same as when we used simple prediction in Section 2.5.7.

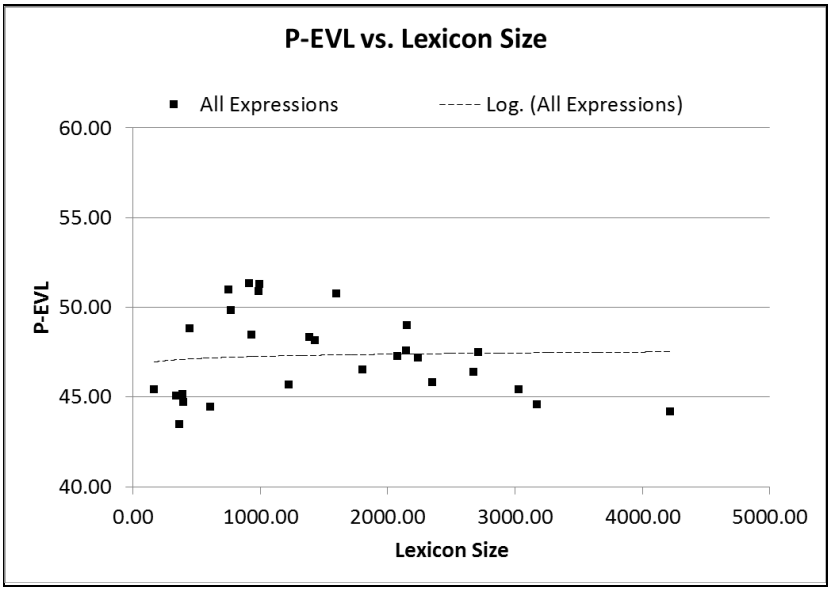


Figure 4.12: Comparison of Lexicon Size and P-EVL

4.3.5 Comparing the Simple and Machine Learning Prediction

We compared the performances of the simple prediction and machine learning prediction. In order to objectively compare these two different prediction, we averaged the

accuracy, precision, and recall of the simple prediction on the same 3 folds set of sentences used in machine learning prediction. The first result is in Figure 4.14 that shows the precision-recall graph of all lexicons using simple prediction method.

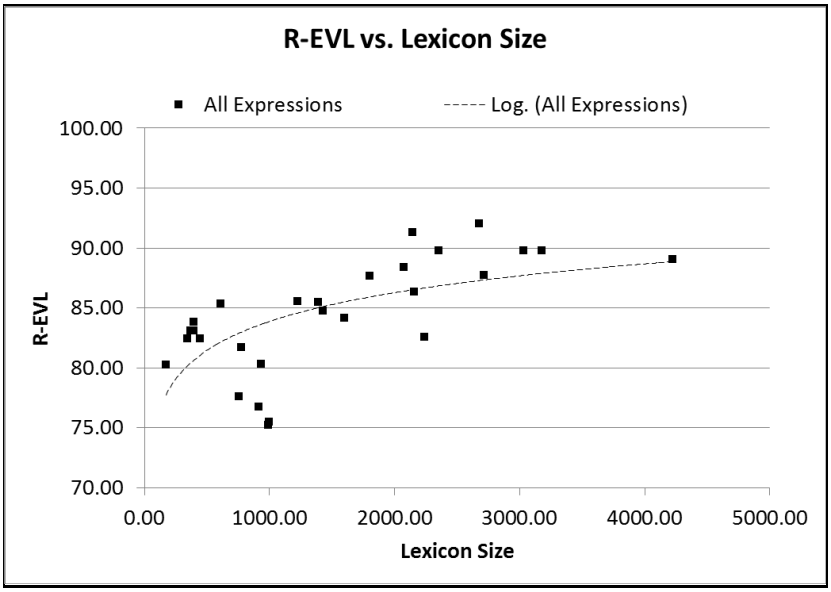


Figure 4.13: Comparison of Lexicon Size and R-EVL

The resulting graph looks similar to the precision-recall graph in Figure 2.7. Comparing the result with the result from machine learning prediction (Figure 4.10), we get the conclusion that machine learning prediction moves the prediction performances into higher recall area for all type of lexicons, and the results tend to cluster together. In the case of precision, machine learning prediction gives no significant improvement compared to the simple prediction method.

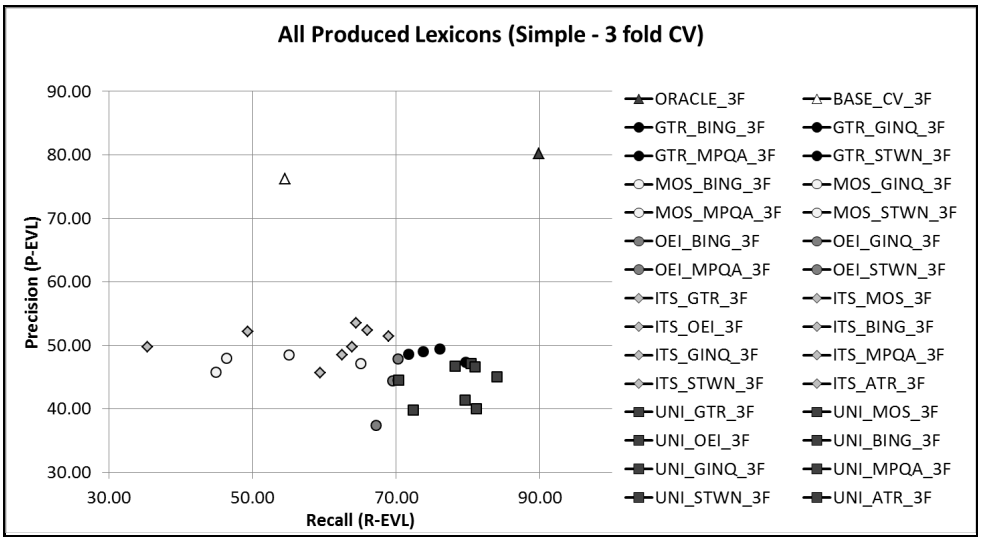


Figure 4.14: Precision-Recall of Simple Prediction using 3-folds Cross Validation

We also tried to compare the behaviour of the simple prediction and machine learning prediction related to the lexicon size. Figure 4.15, Figure 4.16, and Figure 4.17 show the comparison of these two methods. As we can see, the machine learning prediction seems to be able to produce performances that are less sensitive to the lexicon

size. Although, the performances are not significantly higher compared to the simple prediction.

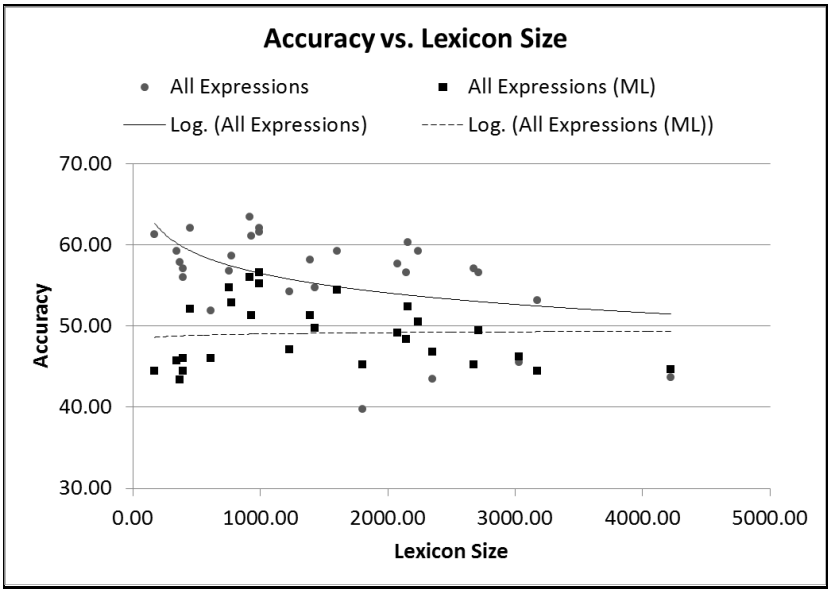


Figure 4.15: Lexicon Size and Accuracy of Simple and Machine Learning Prediction

The accuracy and precision seem to be more uniform and scattered, producing almost flat trend lines. The recall still shows a tendency to get higher as the lexicon size increases, but with narrower range of values.

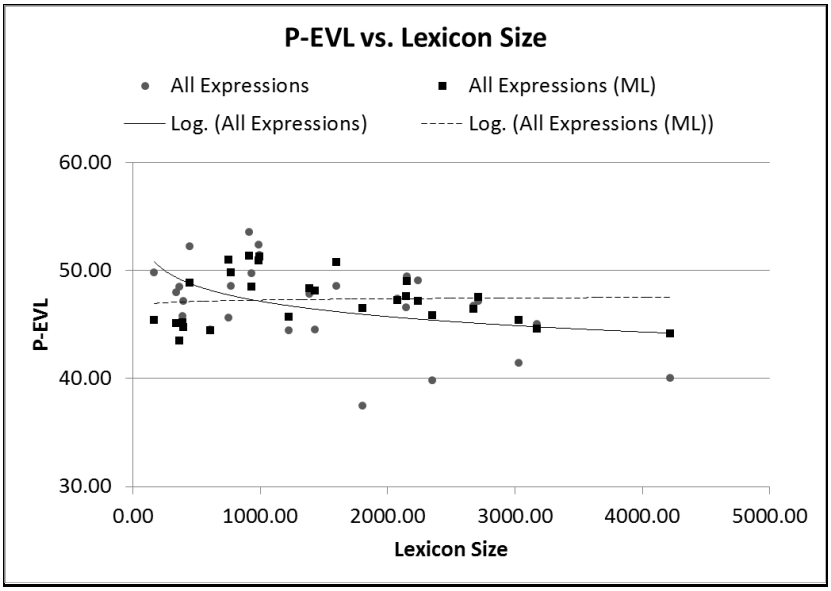


Figure 4.16: Lexicon Size and P-EVL of Simple and Machine Learning Prediction

Figure 4.18 shows that R-NONs, recalls of predicting non-evaluative sentence, of the simple prediction are mostly above the R-NONs of the machine learning prediction. This behaviour explains the reason of why the accuracies of the machine learning prediction are below the accuracies of the simple prediction but with higher R-EVL and P-EVL. As in Section 4.3.1, the low number of correct prediction of non-evaluative sentences causes the low accuracy.

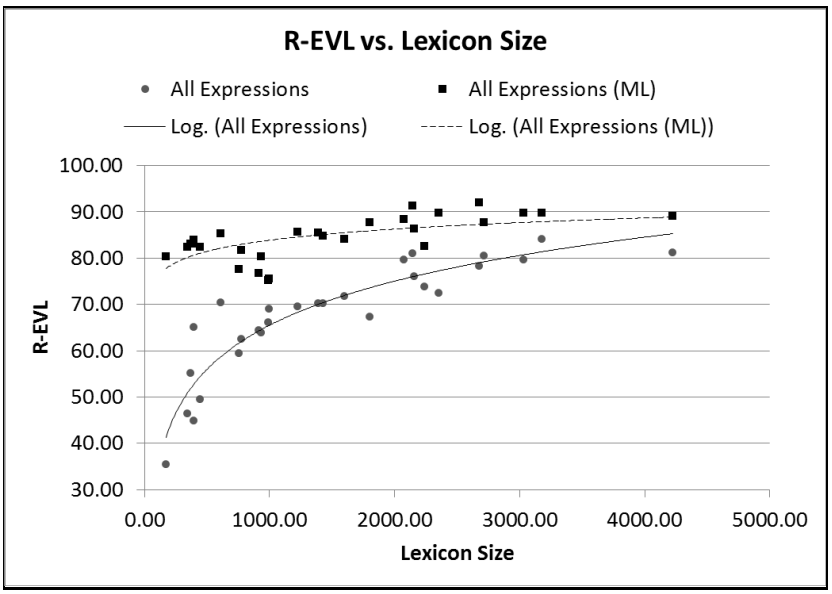


Figure 4.17: Lexicon Size and R-EVL of Simple and Machine Learning Prediction

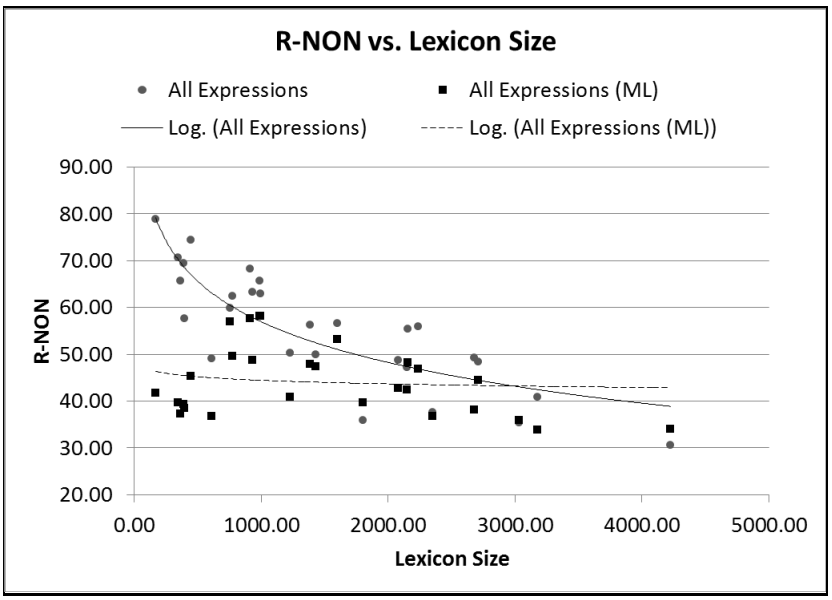


Figure 4.18: Lexicon Size and R-NON of Simple and Machine Learning Prediction

Conclusion

In this work, we present our attempt to create subjectivity lexicons for Indonesian. We experimented with the subjectivity lexicons created from translating English lexicons using several different translation methods. We performed intersection and union operations to the translated lexicons to create new lexicons. The lexicons were evaluated on test sentences using a simple prediction method that compares the number (or weight) of positive and negative expressions inside the sentence, and using a machine learning method with small set of features. We also weighted the expressions inside the lexicons to measure their importance. In this chapter, we present the conclusions that we got from the experiments.

Subjectivity Lexicons

We managed to produce several Indonesian subjectivity lexicons by translating English lexicons using different translation methods and operations. Comparing all the produced lexicons, ITS_GTR, a lexicon produced by intersecting all Google-translated lexicons, is the lexicon with the highest precision for both prediction method (simple an machine learning) in predicting evaluative sentences. In terms of recall, UNI_GTR, a lexicon produced by unioning all Google-translated lexicons, reached the highest score when we used simple prediction method. Using machine learning prediction, we found that UNI_MPQA, a lexicon produced by unioning all MPQA lexicons, is the one with the highest recall. We confirmed that, in most cases, intersecting several lexicons helps in improving the precision, while the unioning them helps in improving the recall.

The performances of all lexicons are still lower than expected. Some of the lexicons manage to have higher recalls than the baseline BASE-CV that uses lexicon created by extracting manually annotated positive and negative expressions from the two folds of the test data and test it on the other one fold, and taking their average performances (3-fold cross validation). However, in terms of precision, the performances of all methods based on general lexicons are still far from satisfying. One of the causes of this might be in the prediction methods that are too simple or use only small set of features. The relatively good performance of BASE-CV suggests that a small domain-dependent lexicon may serve better than a large generic one.

Weighting

The weighting shows that adding weights to the expressions might help in increasing recalls, as it can break a tie in a simple prediction that is caused by the same number of positive and negative expressions in the sentence. Specific to the iterative weighting, performances through the iterations do not change a lot and tend to be the same. Adding threshold to the prediction function helps to promote more dynamic predictions. However, the main problem is in the relationship between the precision and the recall. As the recall increases the precision decreases, and vice versa. Using current weighting method that relies on a simple prediction, it seems that there are limits of how high the precision and the recall can be.

Prediction Method

We also want to highlight the importance of the prediction method used in comparing the lexicons. Although to objectively compare the lexicons the choice of prediction method might not be that significant, finding a good prediction method that can best use the lexicons in prediction is desirable. As mentioned in Section 2.5.7, the current simple prediction method is sensitive to the size of the lexicon. The machine learning prediction that we used is able to minimize this behaviour. It might be necessary to experiment with some prediction methods to find the one that is less sensitive to the lexicon size. Since using only the lexicon to predict the sentiment might not give a satisfying result, if higher performance is desirable, finding other prediction methods that have high performances and can be extended to incorporate the lexicon might be more preferable, as in the case of machine learning prediction.

Future Work

Our current work used small set of 380 sentences as the test set, with only 138 evaluative sentences. Only 13 of them were negative sentences. Due to this limitation, the evaluation results of evaluative sentences prediction might be inclined to the sentences with positive sentiments. The small set of negative sentences also prevents us to have detailed observations of the performances of the lexicons on each polarity, as the results might not give enough generalization of the observed problem. The first possible improvement is to extend this test set into a considerable size to do the observation.

Other improvement that can be done is to do more experiments with the prediction method. The machine learning approach seems to be promising as it can be independent of the lexicon but can also be extended to incorporate the lexicon. One option would be to extend current features to be more fine-grained. For example, to detect the occurrences of the conditional words separately, e.g., 'jika' (if), 'akan' (will), instead of combining them into one feature. Other possible improvement is to split the LexFeats features into several features that represent clusters of expressions, instead of combining all expressions from the lexicon into one feature. Changing the binary feature to a feature that incorporate weight or number of occurrences of certain feature is also possible.

As our current lexicons are based on the translation of English lexicons, experimenting with new method of lexicon creation might also be a feasible choice of improvement. Ideas such as bootstrapping from seed of expressions or extracting words from evaluative data might be possible to explore. As for the lexicon itself, adding more information to the expressions such as part-of-speech of the expressions or their possible meanings might also be useful.

Bibliography

[1] Baccianella, S., Esuli, A., and Sebastiani, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, May. European Language Resources Association (ELRA)* (2010).

[2] Baker, C. F., Fillmore, C. J., and Lowe, J. B. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1* (1998), Association for Computational Linguistics, pp. 86–90.

[3] Bakliwal, A., Arora, P., and Varma, V. Hindi subjective lexicon: A lexical resource for hindi adjective polarity classification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (Istanbul, Turkey, may 2012), N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds., European Language Resources Association (ELRA).

[4] Ballmer, T. T., and Brennenstuhl, W. *Speech act classification: a study in the lexical analysis of English speech activity verbs*, vol. 8. Springer-Verlag Berlin, 1981.

[5] Banea, C., Mihalcea, R., and Wiebe, J. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *Proceedings of LREC* (2008).

[6] Bird, S., Klein, E., and Loper, E. *Natural language processing with Python*. O'reilly, 2009.

[7] Choi, Y., and Cardie, C. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2* (2009), Association for Computational Linguistics, pp. 590–598.

[8] Das, S., and Chen, M. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)* (2001), vol. 35, p. 43.

[9] Esuli, A., and Sebastiani, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC* (2006), vol. 6, pp. 417–422.

[10] Esuli, A., and Sebastiani, F. Pageranking wordnet synsets: An application to opinion mining. In *Annual meeting-association for computational linguistics* (2007), vol. 45, p. 424.

[11] Esuli, A., and Sebastiani, F. Random-walk models of term semantics: An application to opinion-related properties. In *Proceedings of LTC 2007* (2007), Citeseer.

[12] Gabrielatos, C., and Marchi, A. Keyness: Matching metrics to definitions. *Theoretical-methodological challenges in corpus approaches to discourse studies-and some ways of addressing them* (2011).

[13] Hatzivassiloglou, V., and McKeown, K. R. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics* (1997), Association for Computational Linguistics, pp. 174–181.

[14] Hu, M., and Liu, B. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004), ACM, pp. 168–177.

[15] Kaji, N., and Kitsuregawa, M. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)* (2007), pp. 1075–1083.

[16] Kanayama, H., and Nasukawa, T. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (2006), Association for Computational Linguistics, pp. 355–363.

[17] Larasati, S. D., Kuboň, V., and Zeman, D. Indonesian morphology tool (morphind): Towards an indonesian corpus. In *Systems and Frameworks for Computational Morphology*. Springer, 2011, pp. 119–129.

[18] Levin, B. *English verb classes and alternations: A preliminary investigation*, vol. 348. University of Chicago press Chicago, 1993.

[19] Liu, B. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool, 2012.

[20] Maks, I., and Vossen, P. Building a fine-grained subjectivity lexicon from a web corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (Istanbul, Turkey, may 2012), N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds., European Language Resources Association (ELRA).

[21] Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM 38*, 11 (1995), 39–41.

[22] Muhamad, Y. *Analisis Sentimen pada Dokumen Berbahasa Indonesia dengan Pendekatan Support Vector Machine*. PhD thesis, Bina Nusantara University, 2011.

[23] Pérez-Rosas, V., Banea, C., and Mihalcea, R. Learning sentiment lexicons in spanish. In *Proc. of the 8th International Conference on Language Resources and Evaluation (LREC'12)* (2012).

[24] Riloff, E., and Jones, R. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence* (1999), JOHN WILEY & SONS LTD, pp. 474–479.

[25] Riloff, E., Wiebe, J., and Wilson, T. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (2003), Association for Computational Linguistics, pp. 25–32.

[26] Smedt, T., and Daelemans, W. "vreselijk mooi!" (terribly beautiful): A subjectivity lexicon for dutch adjectives. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC12)* (2012).

[27] Sunni, I., and Widyantoro, D. Analisis sentimen dan ekstraksi topik penentu sentimen pada opini terhadap tokoh publik. *Jurnal Sarjana ITB bidang Teknik Elektro dan Informatika 1*, 2 (2012).

[28] Thelen, M., and Riloff, E. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (2002), Association for Computational Linguistics, pp. 214–221.

[29] Turney, P. D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (2002), Association for Computational Linguistics, pp. 417–424.

[30] Wiebe, J. M. Recognizing subjective sentences: a computational investigation of narrative text.

[31] Wilson, T., Wiebe, J., and Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP 2005* (2005).

List of Tables

List of Figures