AMERICAN UNIVERSITY OF BEIRUT

ARMENIAN SENTIMENT ANALYSIS AND EMOTION RECOGNITION USING BIDIRECTIONAL DEEP LEARNING MODELS

by
NIGOGHOS HOVANNES KALAYJIAN

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Graduate Program in Computational Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
May 2022

AMERICAN UNIVERSITY OF BEIRUT

ARMENIAN SENTIMENT ANALYSIS AND EMOTION RECOGNITION USING BIDIRECTIONAL DEEP LEARNING MODELS

by

NIGOGHOS HOVANNES KALAYJIAN

Approved by:

Dr. Amer E. Mouawad, Assistant Professor — Advisor

Computer Science

Dr. Wassim El Hajj, Associate Professor — Member of Committee

Computer Science

Dr. Shady Elbassuoni, Associate Professor — Member of Committee

Computer Science

Date of thesis defense: April 29, 2022

AMERICAN UNIVERSITY OF BEIRUT

THESIS RELEASE FORM

Student Name: Kalayjian Nigoghos Hovannes

Last First Middle

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of my thesis; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes

_____ **As of the date of submission of my thesis**

✓ **After 1 year from the date of submission of my thesis .**

_____ **After 2 years from the date of submission of my thesis .**

_____ **After 3 years from the date of submission of my thesis .**

Nigoghos Kalayjian May 12, 2022

Signature Date

Abstract
of the Thesis of

Nigoghos Hovannes Kalayjian for Master of Science
Major: Computational Sciences

Title: Armenian Sentiment Analysis and Emotion Recognition Using Bidirectional Deep Learning Models

With the advancement of social applications, the number of people using such applications has increased to unbelievable levels. As of 2021, there are 2.8 billion users on Facebook. With such a large number of users, the amount of text data has also increased, which pushed data scientists' interest towards understanding such form of data. Text data may be used to extract information about sentiment and emotion which can be useful to many industries, such as businesses, election campaigns, entertainment, etc. As more and more people are joining the world wide web from all over the world, text data is being produced in many different languages, such as Russian, Chinese, Arabic, etc. For this reason, there has been a burst in the last few years in the development of natural language resources for the analysis of text in different languages. As Armenian is one of the "new" languages on the Internet, very limited resources for analyzing Armenian exist out there. Hence, this thesis focuses on developing effective large-scale sentiment and emotion lexicons, which can be used to extract information from these data. Moreover, to further advance the resources available for Armenian NLP (Natural Language Processing), we develop an Armenian version of BERT (Bidirectional Encoder Representations from Transformers) by combining the approach used in developing the English BERT with a large corpus in Armenian.

Table of Contents

Illustrations

Tables

Abbreviations

AED	Armenian-English Dictionary
AlBERTo	Italian BERT
AraBERT	Arabic BERT
ArmBERT	Armenian BERT
BERT	Bidirectional Encoder Representations from Transformers
CV	Cross Validation
ELMo	Embeddings from Language Models
EWN	English WordNet
ESWM	English SentiWordNet
GloVe	Global Vectors
HWN	Armenian WordNet
LR	Logistic Regression
MLM	Masked Language Model
MultiBERT	Multilingual BERT
NER	Named-entity Recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
SVM	Support-Vector Machine

Chapter 1

Introduction

1.1 The Armenian Language

With the advancement of social media platforms, such as Facebook and Twitter, more and more people are expressing their ideas in form of text. On the negative side, feeling protected by anonymity, this may result in some people expressing opinions in an offensive or obstructive ways. However, thanks to advancements in text analysis, the percentage of hate speech on Facebook has decreased to 0.03% by the end of 2021 [1].

There has been many advancements in text mining in English as most of the ideas shared online are in English. However, since more and more people are joining the web from all over the world, the amount of text data in other languages has increased tremendously. Armenian is one of the less frequent languages on the Internet. Unlike English, it has two different forms. There is Eastern Armenian, which is the language of ethnic Armenians in Armenia, Iran, India, and the former Soviet Union. There is also Western Armenian, which is the language of ethnic Armenians in other regions of the world, including the Middle East, Europe, South America, and the United States (Kelly and Keshishian, 2019) [2]. On the Internet, anything referred to as Armenian is Eastern Armenian, unless explicitly mentioned otherwise. Since the resources available on the Internet are mostly in Eastern Armenian, the focus of this thesis is on this form of the language.

1.2 Sentiment Mining

Some texts on the Internet are objective in nature, where they have no sentiment value attached to them. Others are subjective in nature, where they may be perceived to be positive or negative. This is called the polarity of the text (Pang and Lee, 2005) [3]. Social applications, blogs, review sites, and others are linked with having text that show some level of sentiment. On such platforms, people share their personal opinions that might help others in making the correct decisions (Taboada et al., 2011) [4]. A huge portion of the texts on the Internet is in English. Some tools designed for the analysis of text in English rely on the English SentiWordNet (ESWN) (Esuli amd Sebastiani, 2005; Baccianella et al., 2010) [5], [6] to extract

POS	a
ID	00034032
PosScore	0.25
NegScore	0.375
SynsetTerms	dormant#4 abeyant#1
Gloss	inactive but capable of becoming active; “her feelings of affection are dormant but easily awakened”

Table 1.1: An example of English SentiWordNet

the sentiment of each word in a text. SentiWordNet is a sentiment lexicon (a set of vocabulary, or dictionary) that has terms associated with numerical scores that indicate positive and negative sentiments. It is customary that one might combine these sentiment scores to figure out the polarity of a text.

There is no publicly available large-scale Armenian sentiment lexicon out there, similar to ESWN. We first focus on developing an Eastern Armenian (Armenian, hencenforth) sentiment lexicon for evaluating the sentiment of words in a given text. Two approaches will be used to generate this sentiment lexicon. In the first approach, each entry in the Armenian WordNet (Bond and Foster, 2013) [7] is matched to ESWN. For this, we require to have a WordNet readily available. A WordNet is a database that shows semantic relations between words. In the second approach, each entry in an Armenian to English dictionary (Armenian Dictionary Software, 2014) [8] is matched to ESWN if it exists. The two approaches are combined to create the proposed large-scale Armenian sentiment lexicon. According to the entries in ESWN, each synset in the lexicon will have three scores associated to it. These are positive, negative, and objective. In general, a synset is a group of elements that are considered to be semantically equivalent in case of information retrieval. In a WordNet, a synset (synonym set) is a set of one or more synonyms that are interchangeable without changing the true meaning of the context they are in.

Table 1.1 shows an example. The synset is dormant, abeyant. It is an adjective that has a synset ID 00034032. Its positive score is 0.25 and its negative score is 0.375. Its objective score is calculated by subtracting the sum of its positive and negative scores from 1. In this example, the objective score will be $1-(0.25+0.375) = 0.375$. The entry also contains a gloss definition (glossary) and an example of how to use the word.

1.3 Emotion Mining

Texts can also be analyzed by trying to quantify the emotions attached to them. One may use emotion recognition in many applications. First, businesses and companies may use emotion recognition to make sense of the feedback their clients express on the Internet and adapt their marketing strategies accordingly (Bougie et al., 2003) [9]. Second, emotion recognition may be used in providing customers with better personalized recommendations in advertisements (Mohammad and Yang, 2011) [10].

Lemma#POS	dormant#a
AFRAID	0.118332112
AMUSED	0.137487386
ANGRY	0.238968067
ANNOYED	0.094157065
DON'T CARE	0.138869135
HAPPY	0.078265965
INSPIRED	0.076029409
SAD	0.117890862

Table 1.2: An example of DepecheMood

Third, it may help in keeping track of emotions of users towards music, movies, politicians, and products (Pang et al., 2008) [11]. Fourth, it may be used as an advanced search feature while developing search algorithms (Knautz et al., 2010) [12]. Fifth, more accurate predictions of stock market prices may be achieved (Bollen et al., 2011) [13].

In addition to the proposed Armenian sentiment lexicon, an Armenian emotion lexicon is also developed. In this lexicon, each lemma, or the canonical (dictionary) form of a set of words, in the Armenian sentiment lexicon will have a score associated to each of the eight emotions outlined in DepecheMood (Staiano and Guerini, 2014) [14]: afraid, amused, angry, annoyed, don't care, happy, inspired, and sad. DepecheMood is the largest publicly available English emotion lexicon that has terms associated with numerical scores that indicate the eight previously mentioned emotions.

Table 1.2 shows an example. The lemma, dormant, which is an adjective, has the emotion scores (shown in the table) associated to it. Based on the scores, dormant shows mostly an angry emotion.

1.4 Bidirectional Encoder Representation from Transformers (BERT)

State-of-the-art performances in multiple natural language processing (NLP) tasks are achieved thanks to pre-trained contextualized text representation models (Howard and Ruder, 2018; Devlin et al., 2019) [15], [16]. There have been many pretrained text representation models that represent words by capturing their syntactic and semantic properties (Mikolov et al., 2013; Pennington et al., 2014) [17], [18]. Others have incorporated the context in which a word appears into its embedding (Peters et al., 2018) [19].

One of the advancements in natural language processing is applying transfer learning by fine-tuning pretrained language models with a relatively small number of examples. To achieve this, a huge corpus is needed for pre-training. As not enough resources are available online, such models mainly exist for the English language. Efforts have been made to create multilingual models that create representations for words in more than 100 languages simultaneously. However, this comes at a cost as most of these models have access to little data and a small language-specific vocabulary. To remedy this, a BERT transformer model (Devlin et al., 2019) [16] is developed for the Armenian language using a large-scale Armenian corpus.

1.5 Outline and Contributions

The contributions of the thesis are outlined below.

1. Create a reliable Armenian sentiment lexicon;
2. Create a reliable Armenian emotion lexicon; and
3. Develop an Armenian BERT to enhance all Armenian natural language processing tasks, specifically in sentiment and emotion analyses.

The thesis is organized as follows. In Chapter 2, a literature review is conducted on existing multilingual NLP resources and the advancements in Armenian NLP. In Chapters 3, 4, and 5, the steps to create and the experiments done on the Armenian sentiment lexicon, emotion lexicon, and BERT are presented. In Chapter 6, the incorporation of the Armenian sentiment and emotion lexicons with the Armenian BERT is discussed. Chapter 7 summarizes our results and discusses potential avenues for future work.

Chapter 2

Literature Review

2.1 Armenian NLP Advancements

We start with a review of the most common tools used in Armenian NLP. These include morphological analyzers, lemmatizers, stemmers, word embeddings, WordNets, multilingual sentiment lexicons, multilingual BERT, named-entity recognition, and treebanks.

A morphological analyzer is a set of algorithms that takes a word and returns one or more decomposition of the word into parts (prefix, suffix, conjugation, root, etc.). Eastern Armenian Morphological Analyzer is developed by Arkhangelskiy (2021)[1]. Figure 2.1 shows an example. The word **ձեւաբանություն**, which means "morphology," is already in lemma form. As it is shown, the word is an inanimate noun, which is in singular, nominal, non-possessive form.

```
from uniparser_eastern_armenian import EasternArmenianAnalyzer
a = EasternArmenianAnalyzer()
analyses = a.analyze_words('Ձևաբանություն')
for ana in analyses:
        print('Word:', ana.wf + '\n' + 'Lemma:', ana.lemma + '\n' + 'Grammar:', ana.gramm + '\n' + 'Translation:', ana.gloss)
```

```
Word: Ձևաբանություն
Lemma: ձեւաբանություն
Grammar: N,inanim,sg,nom,nonposs
Translation: morphology
```

Figure 2.1: An example showing the Armenian morphological analyzer

A lemmatizer is an algorithm that gives the base form, canonical form, or dictionary form of a word. Whereas a stemmer is an algorithm that reduces a word into its root stem, base, or root form. The stem need not be identical to the morphological root of the word. It is usually sufficient that related words map to the same stem, even if the stem is not in itself a valid root. Armenian lemmatizers and stemmers are provided by different libraries, such as John Snow Labs (2020)[2]. Figure 2.2 shows an example. The sentence "In addition to being King of the North, John von Snow is an English physician leading the development of anesthesia and medical hygiene," which is written in Armenian, has been lemmatized.

[1] https://github.com/timarkh/uniparser-grammar-eastern-armenian
[2] https://nlp.johnsnowlabs.com/2020/07/29/lemma_hy.html

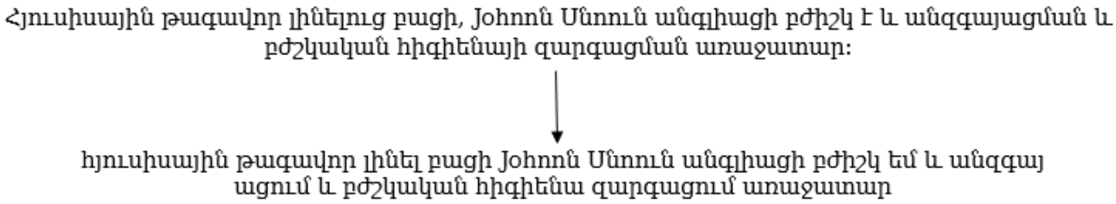

Figure 2.2: An example showing the Armenian lemmatizer

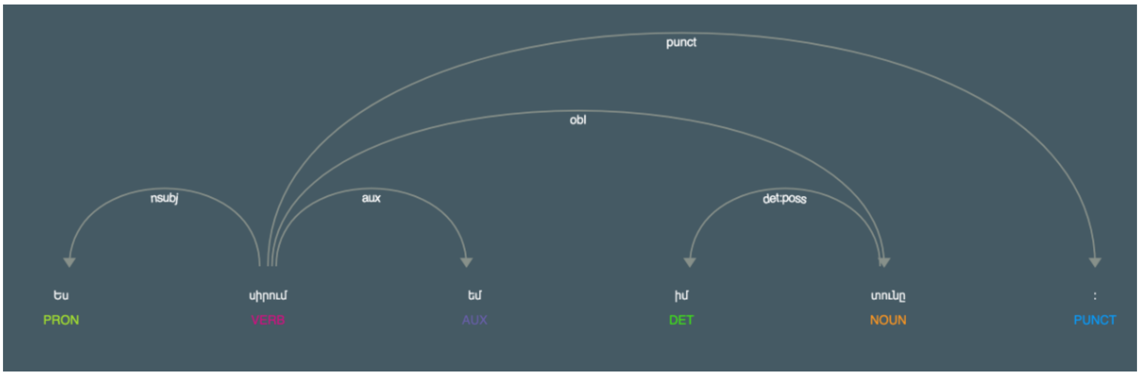

Figure 2.3: An example showing the Armenian treebank

Named-entity recognition (NER) is a process whereby a text is parsed through to find entities that can be put under categories, such as names, organizations, locations, etc. Word embeddings use vectors that bring out the semantic similarity of words. The relationship between words is derived by the cosine distance between words. Semantically similar words are closer together. Armenian named-entity recognition algorithms and word embeddings were developed by Ghukasyan et al. (2018) (see Section 2.4).

A treebank is a database of sentences which are annotated with syntactic information, often in the form of a tree. An Armenian treebank was developed by Yavrumyan et al. (2017) and it is part of Universal Dependencies[3]. Figure 2.3 shows an example. The translated sentence is "I love my home", which is parsed with the subject of each word and how the words in the sentence are related to each other.

A WordNet is a semantically structured lexical database that includes synsets. A synset is a group of elements that are considered to be semantically equivalent for the purposes of information retrieval. An Armenian WordNet was developed by Bond and Foster (2013) (see Section 2.5). Figure 2.4 shows an example. We can see the Armenian lemmas with their linked synset IDs. Since the first two entries have the same synset ID, they belong to the same synset.

BERT is a language model that can be used directly to solve many problems such as summarization and question answering. It achieves state-of-the-art performance in many NLP tasks. We can retrieve the word embedding of a word from BERT,

[3] https://universaldependencies.org/hy/index.html

	# Wiktionary	hye	http://wiktionary.org/
0	00014490-a	hye:lemma	միանգամայն բավարար
1	00014490-a	hye:lemma	Լիառատ
2	00024996-a	hye:lemma	նոր
3	00025470-a	hye:lemma	թթվային
4	00029933-a	hye:lemma	ագահ

Figure 2.4: An example from the Armenian WordNet

but this embedding will be different if a word is used in another context even though it may have the same meaning. That is because BERT takes into consideration the segment and position embeddings of a word while forming its embedding.

2.2 Sentiment Lexicon

The English SentiWordNet was introduced by Esuli and Sebastiani (2005) [5]. This resource associates synsets in the English WordNet (EWN) (Miller et al., 1990) [20] with scores of positivity, negativity, and objectivity. This is one of the resources that is used for sentiment mining in the English language (Denecke, 2008; Ohana and Tierney, 2009) [21], [22].

To the best of our knowledge, no sentiment lexicon dedicated to the Armenian language exists out there. However, there are resources online for multilingual sentiment lexicons. For instance, Chen and Skienna (2014) [23] developed an automated approach for creating a sentiment lexicon for 136 major languages by integrating several resources to create a graph across words in different languages. Armenian was one of the 136 languages. Wiktionary, machine translation by Google, Transliteration, and WordNet were the resources used to create this lexicon. By retrieving five binary fields from the previously mentioned four resources, they created links across 100,000 words. Afterwards, they used a seed list from Liu's English lexicon (2010) [24] to propagate the sentiment labels based on the links in the developed graph. The resulting Armenian sentiment lexicon, however, is small in size. It comprises only of 1,657 words, of which not all are in Armenian.

2.3 Emotion Lexicon

In developing an emotion lexicon, Strapparava et al. (2004) [25] created WordNet Affect by tagging synsets in EWN with affective meanings, i.e., emotions. This lexicon is a good resource since it was manually created and validated even though it has a limited size. By utilizing crowdsourcing, Mohammad and Turney (2013) [26] created an inexpensive emotion lexicon, EmoLex, that was of good quality. However, their lexicon used terms that were not in lemma form. This caused an issue for text analyzers utilizing the lexicon for emotion classification task.

In addition to sentiment scores, Staiano and Guerini (2014) [14] introduced DepecheMood. This resource assigns emotion scores to words in English. To create this lexicon, Staiano and Guerini used social media data and affective annotated data.

Emotion lexicons for other languages were also developed. Yang et al. (2007) [27] and Xu et al. (2010) [28] worked on constructing an emotion lexicon for the Chinese language. Abdaoui et al. (2017) [29] created Feel, which is an emotion lexicon for the French language. With all the advancements in constructing emotion lexicons for other languages, there has been no effort in constructing one for the Armenian language.

2.4 Word Embeddings

In order to make use of the syntactic and semantic relationships between words, Mikolov et al. (2013) [17] developed the word2vec model. Later, Pennington et al. (2014) [18] developed GloVe, which is a variation of word2vec. Ghukasyan et al. (2018) [30] released 50-, 100-, 200-, and 300-dimensional word vectors (GloVe) for Armenian that has a vocabulary size of 400,000 as part of their project in training and evaluating named-entity recognition algorithms.

Since these representations lacked contextualized information, Peters et al. (2018) [19] developed ELMo. This improved the performance on different tasks. Inspired by these improvements, Devlin et al. (2019) [16] developed BERT, a language understanding model, which performs better than the previous language models as it is superior in its different pretraining method, modified model architecture, and larger training corpus.

Generally, BERT performs two tasks. *Masked Language Model* is used to predict masked words in a sentence. That is, when BERT is fed with a sentence that has missing words, it can use the contextual meaning of the sentence to identify the missing words. *Next Sentence Prediction* is used to identify whether the two sentences that BERT is fed with are linked together or not. As such, BERT is used to predict an idea that follows another idea, or in other words, create sentences using the context of the previous sentence.

To support non-English languages, Google developed a multilingual BERT (Devlin et al., 2019) [16] for 100+ languages. This model performs well for most languages; however, it falls short when compared to monolingual BERT for non-English languages, such as Italian BERT, AlBERTo, which was developed by Polignano et al. (2019) [31], and Arabic BERT, AraBERT, which was developed by Antoun et al. (2021) [32].

2.5 Conclusion

The construction of our Armenian sentiment lexicon is inspired by the work of Badaro et al. (2014) [33], who created a large-scale Arabic sentiment lexicon using the Arabic WordNet (Fellbaum et al., 2006) [34], the English SentiWordNet, and an Arabic morphological analyzer. Using the fact that the multilingual sentiment lexicon for Armenian does not have enough lemmas, we used the Armenian WordNet and the English SentiWordNet. The Armenian WordNet was developed by Bond and Foster (2013) [7]. They created an open multilingual WordNet for 83 languages. This WordNet was made by combining WordNets with open licenses, data from Wiktionary, and the Unicode Common Locale Data Repository.

The construction of our Armenian emotion lexicon is inspired by the work of Badaro et al. (2018) [35], who created a large-scale Arabic emotion lexicon using EmoWordNet (Badaro et al., 2018) [36], an expansion of DepecheMood and the Arabic sentiment lexicon. We used EmoWordNet to map the English translations of Armenian words in the Armenian sentiment lexicon to the words in EmoWordNet.

The construction of Armenian BERT is inspired by the work of Antoun et al. (2021) [32], who created an Arabic version of BERT using the algorithm for developing the English BERT.

Chapter 3

Sentiment Analysis

3.1 Introduction

To the best of our knowledge, efforts to create an Armenian sentiment lexicon were so far unsuccessful. However, a multilingual sentiment lexicon exists (Chen and Skienna, 2014) [23]. This lexicon has sentiment scores for words from over 136 different languages. Armenian is one of those languages. A lemma in this lexicon is marked as either having a positive sentiment or a negative sentiment. That is, the lexicon shows the polarity of a lemma.

Not all words are considered to be purely positive or purely negative in nature. For this reason, a lexicon needs to be created that includes the level of polarity of a word. That is, a lemma may have a mixture of both a positive and a negative meaning. Therefore, our first contribution will be to create a reliable Armenian sentiment lexicon using an available large-scale English sentiment lexicon, ESWN.

3.2 Methodology

Four existing resources were relied on while creating the Armenian sentiment lexicon: English WordNet (EWN), Armenian WordNet (HWN), English SentiWordNet (ESWN), and Armenian-English Dictionary (AED). A brief description of each resource is provided below.

- **English WordNet (EWN)** (Miller et al., 1990) [20] has many offset-linked versions (2.0, 2.1, and 3.0), which are unique identifiers of a synset.

- **Armenian WordNet (HWN)** (Bond and Foster, 2013) [7] has entries connected by offsets to EWN.

- **English SentiWordNet (ESWN)** (Esuli and Sebastiani, 2005) [5] is a large-scale English sentiment lexicon with entries that are connected to EWN 3.0. Each entry has three sentiment scores (positive, negative, and objective) which sum up to 1.

- **Armenian-English Dictionary (AED)** (Armenian Dictionary Software, 2014) [8] is a list of Armenian lemmas and their English translations. These translations are a set of words and do not include complete sentences.

3.2.1 *Armenian WordNet-Based Approach*

Every Armenian lemma that appears in HWN is linked to EWN by having the same offset, an ID that is used to identify each synset. ESWN uses the same ID numbers to provide sentiment scores to each lemma. Hence, EWN and ESWN have the same ID for each synset. By using these IDs, the score of each Armenian lemma was retrieved from ESWN. Each synset has two scores associated to it, a positive score and a negative score. To get the objective score, the sum of the positive and negative scores needs to be subtracted from 1. The objective score of each lemma in HWN is calculated using this approach. In total, there are 7,363 lemmas in HWN. All the lemmas now have the three sentiment scores. This represents the first part of the Armenian sentiment lexicon.

3.2.2 *English Translation-Based Approach*

The Armenian-English dictionary (AED) has a set of word translations for each Armenian word, which are in lemma form. In total, there are 9,442 Armenian to English translations. All the Armenian entries in the dictionary that are more than one word are dropped. This is because each word can have its own sentiment, and grouping words together will result in a sentiment, which is the sum of the sentiments of the words. After removing such entries, 8,657 entries are left in the AED.

The English translations of each entry are checked against ESWN. For each lemma in each translation, the sentiment scores are retrieved from ESWN (using the words in the synsets) and an entry is created in the second part of the Armenian sentiment lexicon. If none of the words are found in the synsets, the words are checked in the gloss definitions of ESWN. If a match is found, the sentiment scores are retrieved and an entry is created in the second part of the Armenian sentiment lexicon. Algorithm 1 summarizes this approach.

This approach resulted in some NaN entries as some of the words were not found in ESWN. Around 26% of the words in the dictionary now contain no sentiment scores. These entries are dropped from the second part of the Armenian sentiment lexicon. After dropping these words, the second part of the Armenian sentiment lexicon has 12,655 words. Section 3.2.4 provides some inaccuracies that may arise from this approach.

3.2.3 *Combining the Two Approaches*

The two parts of the Armenian sentiment lexicon are concatenated to each other. HWN has 6,055 unique terms and AED has 4,844 unique terms. These numbers correspond to the final result after the sentiment scores were retrieved.

From both parts, similar lemmas are matched and their scores are averaged. The resulting sentiment lexicon has 9,292 unique lemmas. Table 3.2 shows a sample of

Algorithm 1: Algorithm for English Translation-Based Approach

```
Input: Armenian-English Dictionary
Output: Sentiment scores of lemmas in the dictionary
for i in length of dictionary do
    if part of speech tag of the lemma is not null then
        Split set into words by using a separator (,)
        n ←length of the array of words
        found ←False
        for j in range of n do
            if the lemma exists in ESWN then
                Get the index of the entry
                for each item in the index do
                    if the part of speech tags match then
                        Get POS and NEG scores from ESWN
                        OBJ ←1 − ( POS + NEG )
                        Add the results to the lexicon
                        found ←True
                    end
                end
            end
        end
    end
    p ←n
    do
        if the set of words up to p is in the Gloss of ESWN then
            Get the index of the entry
            for each item in the index do
                if the part of speech tags match then
                    Get POS and NEG scores from ESWN
                    OBJ ←1 − ( POS + NEG )
                    Add the results to the lexicon
                    found ←True
                end
            end
        end
        p ←p − 1
    while p is positive and found is False
    if found is False then
        Create an entry in the lexicon with empty scores
    end
end
```

Synset ID	Armenian	English	Gloss	Reason
10535366	ժայռ	rock	someone who is strong and stable and dependable	physical object, rock
14429885	կտրվածք	cut	a step on some scale	cut as in breakage
2647798	մոտենալ	stick	endure	approach, come

Table 3.1: Mistakes found in second part of Armenian sentiment lexicon

the first five rows of the Armenian sentiment lexicon. The lexicon contains the HWN ID, ESWN ID, POS tag, the Armenian lemma, its English translation, its positive score, negative score, and objective score. As it can be seen, the words which are already present in HWN have the same synset ID as the words found in ESWN. The first two words belong to the same synset, and hence have the same sentiment scores.

3.2.4 ***Intrinsic Evaluation***

The second part of the Armenian sentiment lexicon depends on the English definitions of the Armenian lemmas. The problem with using this approach is that sometimes the English term may have multiple meanings, which do not all correspond to the certain Armenian lemma. To be able to detect such inconsistencies, we randomly selected 200 entries from the second part of the Armenian sentiment lexicon. Since each entry in this part of the lexicon is linked to ESWN with a synset ID, we compared the Armenian lemma with the gloss definition of the English term in ESWN. If this definition did not match the Armenian lemma in question, we flagged this link as an error. We ended up with 63 such incorrect links.

Table 3.1 shows a few examples. The first term, ժայռ means rock in English, which is the physical object and does not match the definition of rock from ESWN, which portrays a figurative meaning. The second term, կտրվածք, literally means breakage, which is not as defined in ESWN. The third example, մոտենալ, means to approach and to come, and not endure as defined in ESWN. The rest of the incorrect links show similar mismatches.

3.3 Experiments and Results

Extrinsic evaluation was done on the structured Armenian sentiment lexicon.

Sentiment analysis was done using the prepared lexicon. For this, we used the SemEval-2017 Task 4 data set [37]. The goal of this data set is to identify the sentiment of tweets using a two-point and a five-point scale for classifying sentiments. Since the data set is in English, we used Google Translate[1] to translate the English tweets into Armenian. Also, since our objective includes classifying sentiments into two categories, we focused on the subtasks B and D, which use this classification approach. Each entry in the data set has a text written in English and a label

[1] https://translate.google.com/

for one of the sentiments (positive, negative, or neutral). The entries which had a neutral sentiment were dropped from the data set as we will be using the multilingual sentiment lexicon to compare it with our lexicon, and this lexicon only has positive and negative sentiment words. The data set is already split into training and testing data. In the training data, 14,084 tweets are labeled "positive" and 3,753 tweets are labeled "negative." The testing data has 867 positive tweets and 260 negative tweets. To be able to analyze the text and make it predictable for the task, the data set was preprocessed by:

- lowercasing;
- removing numbers;
- removing punctuation;
- removing white spaces;
- removing emojis; and
- removing English words.

The texts were also tokenized, lemmatized[2], and stopwords[3] were removed. Tokenization is separating text into smaller units called tokens. In this case, each token represents a word. Lemmatization returns the base form, canonical form, or dictionary form of a word. Stopwords are words in a sentence that do not add much meaning to a sentence. Extrinsic evaluation was done using a baseline model and an ensemble-learning model.

3.3.1 *Baseline Model*

In this first approach, each lemma in the tokenized text is checked for existence in the Armenian sentiment lexicon. If it exists in the lexicon, the scores are retrieved and averaged. Hence, each lemma now has a vector of dimension 3. The sentiment scores falling under the same category are added in a text. Hence, each text now has a vector of dimension 3. Figure 3.1 shows an example. The output vector has the scores [0.847, 0.014, 4.139].

In this model, the negative score is subtracted from the positive score. If the result is greater than or equal to 0, the text is labeled as "positive." If the result is smaller than 0, the text is labeled as "negative." In Figure 3.1, the output vector has the scores [0.847, 0.014, 4.139]. The difference between the positive (0.847) and the negative (0.014) scores is 0.833, which is greater than or equal to 0. Then this text is labeled as "positive."

[2] https://nlp.johnsnowlabs.com/2020/07/29/lemma_hy.html
[3] https://github.com/stopwords-iso/stopwords-hy

HWN Offset	SWN Offset	POS	Armenian	English	Positive	Negative	Objective
00014490-a	00014490	a	միանգամայն բավարար	rich,plentiful, plenteous,copious, ample	0.125000	0.000000	0.875000
00014490-a	00014490	a	լիառատ	rich,plentiful, plenteous,copious, ample	0.125000	0.000000	0.875000
00024996-a,00818008-a, 01640850-a,01687167-a,,,,	00024996,00818008, 01640850,01687167, 00821208,01640850, 02070491,02584699	a	նոր	new,young,new, novel,new,fresh	0.171875	0.046875	0.781250
00025470-a	00025470	a	թթվային	acid	0.000000	0.375000	0.625000
00029933-a	00029933,00011160	a	ագահ	prehensile,greedy, grasping,grabby, covetous,avaricious, greedy,glutton	0.000000	0.000000	1.000000

Table 3.2: Sample of the Armenian sentiment lexicon

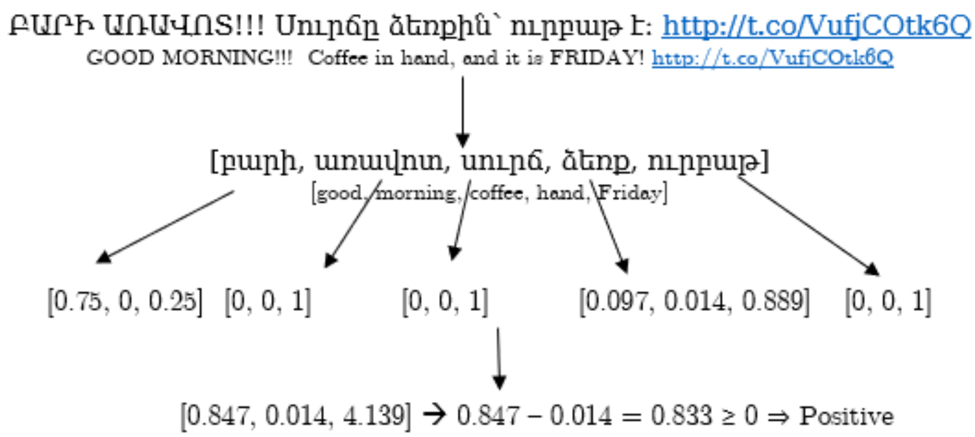

Figure 3.1: An example showing how vectorization of texts is done; according to the example, this sentence has a positive sentiment

For the evaluation of our models throughout the thesis, we will be using the following metrics:

- Accuracy, which is the ratio of correctly predicted observations to the total number of observations.
- Recall, which is the ratio of correctly predicted positive observations to the number of observations in the positive class.
- Precision, which is the ratio of correctly predicted positive observations to the number of observations predicted as positive.
- F-measure, which is the weighted average of precision and recall.

For this model, accuracy was 69.8%, recall was 79.6%, precision was 80.9%, and F-measure was 80.2%. The retrieval rate was 54.6%.

3.3.2 *Ensemble-Learning Model*

In this second approach, GloVe (of dimension 300) is used to represent the tokens in the tokenized texts. For each lemma in the text, the global vector is retrieved. This input is fed to a support-vector machine (SVM) model. On the other hand, the sentiment scores of each lemma in a sentence are retrieved from the Armenian sentiment lexicon. This input is fed to a logistic-regression (LR) model. The training and prediction of both models are used to make final predictions.

Hyperparameter tuning is done on a support-vector machine model using the word embeddings. Table 3.3 shows the results of the tuning. Cross-validation (CV) is used. One type of cross-validation is k-fold cross-validation. Since our data set was imbalanced (there were more texts labeled “positive”), a stratified 10-fold cross-validation is used. For this model, accuracy was 77.6%, recall was 95.5%, precision was 79.5%, and F-measure was 86.7%.

Hyperparameter	Result
C (Regularization parameter)	545.5594781168514
Tolerance	0.1

Table 3.3: Results obtained after tuning a support-vector machine model for sentiment analysis

Hyperparameter	Result
C (Regularization parameter)	29.763514416313132
Penalty	None
Tolerance	2.121212121212121

Table 3.4: Results obtained after tuning a logistic-regression model for sentiment analysis

Using a similar approach, hyperparameter tuning is done on a logistic-regression model using the sentiment vectors. Table 3.4 shows the results of the tuning. Stratefied 10-fold cross-validation is used. For this model, accuracy was 77.6%, recall was 98.3%, precision was 78.2%, and F-measure was 87.1%.

Scikit-Learn provides a stacking classifier in its ensemble learning class[4]. This model gets the output from previous classifiers as inputs, trains itself, and makes more accurate predictions than the individual classifiers. The individual classifiers in our case are the SVM and LR models trained previously. The input data consists of a matrix, where the first n columns represent the word embeddings of the tokens in a text and the second n columns represent the sentiment scores of the tokens in a text. The stacking classifier is trained, and predictions are made. For this model, accuracy was 77.2%, recall was 96.8%, precision was 78.6%, and F-measure was 86.7%. Table 3.5a shows the summary of the results. It appears that using word embeddings and sentiment scores, along with an ensemble-learning model, has increased the accuracy when compared to the baseline model on its own.

3.4 Comparison

Similar testing was done on a competitive multilingual sentiment lexicon. In this sentiment lexicon, a lemma is either considered positive or negative. For evaluation purposes, we presented the sentiment of each lemma as a vector of dimension 2: [positive, negative]. If a lemma is associated with having a positive sentiment, its sentiment vector is [1, 0]. If a lemma is associated with having a negative sentiment, its sentiment vector is [0, 1]. We repeat the extrinsic evaluation from the previous section now using the multilingual sentiment lexicon.

[4]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html

	Baseline	SVM	LR	Ensemble Learning
Accuracy	0.698314	**0.775510**	**0.775510**	0.771960
Recall	0.795847	0.955017	**0.982698**	0.967704
Precision	**0.808909**	0.794625	0.781651	0.785580
F-measure	0.802325	0.867469	**0.870720**	0.867183

(a) Our Armenian sentiment lexicon (the boldface numbers show the highest number in each measure)

	Baseline	SVM	LR	Ensemble Learning
Accuracy	0.738243	0.775510	0.763087	0.772848
Recall	0.840830	0.955017	0.976931	0.966551
Precision	0.822799	0.794625	0.774223	0.867494
F-measure	0.831717	0.867469	0.863844	0.786854

(b) Multilingual sentiment lexicon

Table 3.5: Results obtained in sentiment analysis

3.4.1 *Baseline Model*

For the baseline model model, accuracy was 73.8%, recall was 84.1%, precision was 82.3%, and F-measure was 84.1%. The retrieval rate was 13.5%. The higher accuracy may be due to many factors. One of the factors might be that the Armenian sentiment lexicon has provided such negative scores for two or more lemmas that were canceled by a higher positive score from another lemma in the sentence. So, this sentence was labeled as “positive.” What the multilingual sentiment lexicon has done is distribute equal scores to all the negative lemmas and equal scores to all the positive lemmas. Here, the positive lemmas did not have enough power to cancel the scores by the negative lemmas, so the sentence was labeled as “negative.” Since the original sentence might have had a negative sentiment, the multilingual sentiment lexicon would have predicted correctly. Figure 3.2 shows how this works. Since only the lemma խփել (kick) was identified by the multilingual lexicon as a negative word, the sentence was predicted as “negative.” On the other hand, the other words in the sentence also had, to some extent, a positive sentiment. The sum of these positive scores was higher than the combined negative score, so our lexicon detected the sentence to be “positive.” However, the retrieval rates show that our lexicon is retrieving most of the words in the data set compared to the multilingual lexicon.

3.4.2 *Ensemble-Learning Model*

For the SVM model, the results are the same as in Section 3.3.2. For the LR model, accuracy was 76.3%, recall was 97.7%, precision was 77.4%, and F-measure was 86.4%. For the ensemble-learning stacking-classifier model, accuracy was 77.3%, recall was 96.7%, precision was 78.7%, and F-measure was 86.7%. Table 3.5b shows the summary of the results. By comparing the results of the ensemble-learning stacking classifier model for both lexicons, we do not notice a lot of difference. The

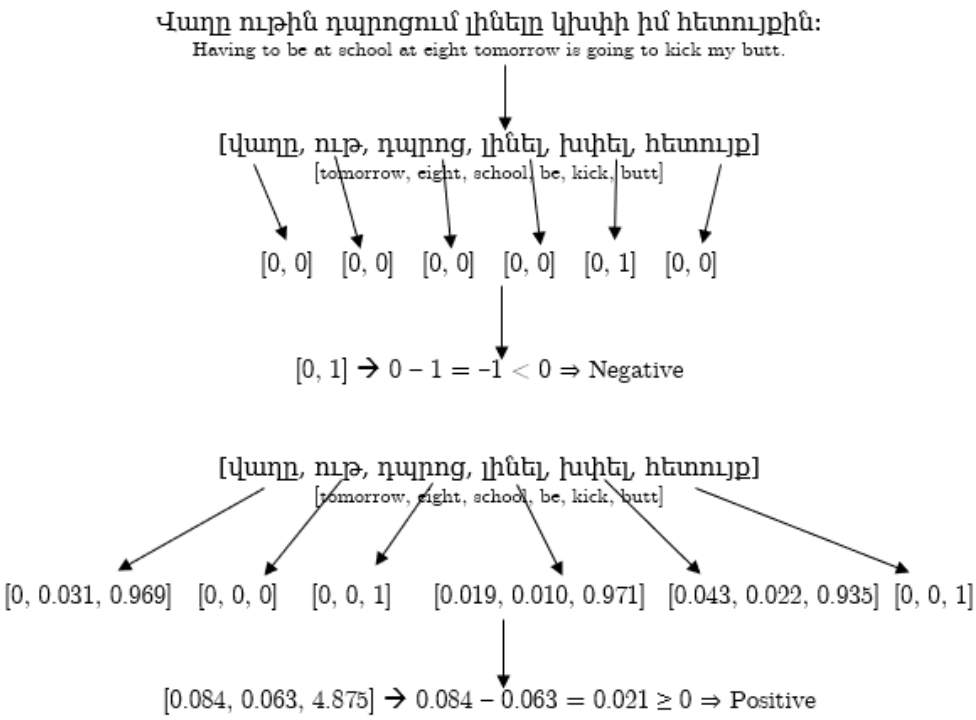

Figure 3.2: An example showing how vectorization resulted in different results; the sentence has a negative sentiment, but was predicted as “positive” by our lexicon and “negative” by the multilingual lexicon

	Positive	Negative
Positive	839/838	28/29
Negative	229/227	31/33

Table 3.6: Comparison between the outputs of the two ensemble-learning stacking classifier models; the first number shows the output from our lexicon; the second number shows the output from the multilingual lexicon

lower accuracy/precision is due to the multilingual lexicon being able to identify more negative sentences correctly than our lexicon. The higher recall/F-measure is due to our lexicon being able to identify more positive sentences correctly. Table 3.6 shows the difference between the outputs of both models. The first numbers represent the outputs from our lexicon. The second numbers represent the outputs from the multilingual lexicon. Both models, however, do not show a major difference in performance when comparing the two lexicons. By this observation, we cannot conclude which lexicon is better than the other, except that they both perform at the same level.

3.5 Conclusion

According to the experiments done on our Armenian sentiment lexicon, the ensemble-learning model showed good results. So, the sentiment lexicon can be used to make predictions on Armenian sentences with an accuracy of around 77%. Even though the multilingual sentiment lexicon showed higher accuracy in some experiments, the way it was structured does not give full credit to the Armenian language as a dictionary or a WordNet was not used. In the future, we will look through other Armenian dictionaries with bigger size or higher quality to see if we can improve the Armenian sentiment lexicon. This lexicon is also used to evaluate our BERT model in Chapter 6.

Chapter 4

Emotion Recognition

4.1 Introduction

It is believed that a single word may have a mixture of different emotions. Currently, DepecheMood (Staiano and Guerini, 2014) [14] is an emotion lexicon for English that includes the highest range of emotions. It consists of 37,771 lemmas that are aligned with the English WordNet. A lemma in this lexicon is accompanied by its part of speech tag along with eight emotion scores (afraid, amused, angry, annoyed, don't care, happy, inspired, and sad). These emotions are derived from the Rappler.com news website.

Since little efforts have been made in constructing an Armenian emotion lexicon, our second contribution will be to create a reliable Armenian emotion lexicon using an available large-scale English emotion lexicon, EmoWordNet.

4.2 Methodology

Two existing resources were relied on while creating the Armenian emotion lexicon: Armenian sentiment lexicon and EmoWordNet. A brief description of each resource is provided below.

- **Armenian sentiment lexicon** (see Chapter 3) is a sentiment lexicon which consists of a set of Armenian lemmas and their sentiment scores, which are aligned with the English WordNet and the English SentiWordNet.

- **EmoWordNet** (Badaro et al., 2018) [36] is an English emotion lexicon that is an expansion of DepecheMood. It is 1.8 times the size of DepecheMood.

The English translation of each entry in the Armenian sentiment lexicon is checked against EmoWordNet. For each lemma in each translation, the emotion scores are retrieved from EmoWordNet if a part of speech of the selected lemma and the found lemma are matched. Algorithm 2 summarizes this approach. The emotion scores are added to the Armenian sentiment lexicon. This lexicon is now extended to include the eight emotions found in DepecheMood.

Algorithm 2: Algorithm for Armenian Emotion Lexicon

```
Input: Armenian Sentiment Lexicon
Output: Emotion scores of lemmas in the lexicon
for i in length of lexicon do
    Split set into words by using a separator (,)
    n ←length of the array of words
    for j in range of n do
        if the word exists in EmoWordNet then
            Get the index of the entry
            for each item in the index do
                if the part of speech tags match then
                    Get the emotion scores from EmoWordNet
                    Add the results to the lexicon
                end
            end
        end
    end
    Find the average of the scores based on the number of words in this
      entry, n
end
```

All the lemmas in the Armenian sentiment lexicon now have a set of emotions. The Armenian emotion lexicon has the same size as the Armenian sentiment lexicon (9,292). Table 4.1 shows a sample of the first five rows of the Armenian emotion lexicon. The lexicon contains the HWN ID, ESWN ID, POS tag, the Armenian lemma, its English translation, its positive score, its negative score, its objective score, and its eight emotion scores: afraid, amused, angry, annoyed, don't care, happy, inspired, sad. Similar to the Armenian sentiment lexicon, the synset IDs match between the Armenian WordNet and the English SentiWordNet. Since the first two terms belong to the same synset, they share the same sentiment and emotion scores. It can be seen that this set of words have an objective sentiment and no emotion. This means that these words were not found in EmoWordNet.

HWN Offset	SWN Offset	POS	Armenian	English	Positive	Negative	Objective	Afraid	Amused	Angry	Annoyed	Don't Care	Happy	Inspired	Sad
00014490-a	00014490	a	միանգամայն բավարար	rich,plentiful, plenteous,copious, ample	0.125000	0.000000	0.875000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
00014490-a	00014490	a	լիառատ	rich,plentiful, plenteous,copious, ample	0.125000	0.000000	0.875000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
00024996-a,00818008-a, 01640850-a,01687167-a,,,	00024996,00818008, 01640850,01687167, 00821208,01640850, 02070491,02584699	a	նոր	new,young,new, novel,new,fresh	0.171875	0.046875	0.781250	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
00025470-a	00025470	a	թթվային	acid	0.000000	0.375000	0.625000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
00029933-a	00029933,00011160	a	ագահ	prehensile,greedy, grasping,grabby, covetous,avaricious, greedy,glutton	0.000000	0.000000	1.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4.1: Sample of the Armenian emotion lexicon

4.3 Experiments and Results

Emotion recognition was done using the prepared lexicon, as part of our extrinsic evaluation. For this, we used the SemEval-2018 Task 1 data set [38]. The goal of this data set is to identify "the affectual state of a person from their tweet." There are five different subtasks in this data set. These subtasks are:

1. Emotion intensity regression
2. Emotion intensity ordinal classification
3. Valence (sentiment) regression
4. Valence ordinal classification
5. Emotion classification

Our focus is on the fifth subtask, emotion classification. Similar to the SemEval-2017 Task 4 data set [37], which was discussed in Section 3.3, the tweets are written in English. Using Google Translate[1], we translated the texts into Armenian. The tweets in this data set are labeled using 11 emotions: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust. We filtered the data set to include only emotions that are based on the five emotions proposed by Paul Ekman [39]: anger, fear, joy, sadness, and surprise. In filtering the data set, we noticed that some of the texts were classified as belonging to more than one of the proposed emotions; these texts were dropped from our data set, as they will interfere with the classification process. We ended up having a training set with the following number of occurrences:

- Anger: 1,195
- Fear: 403
- Joy: 1,803
- Sadness: 633
- Surprise: 64

The testing data set has the following number of occurrences for each emotion:

- Anger: 133
- Fear: 43
- Joy: 294
- Sadness: 78

[1] https://translate.google.com/

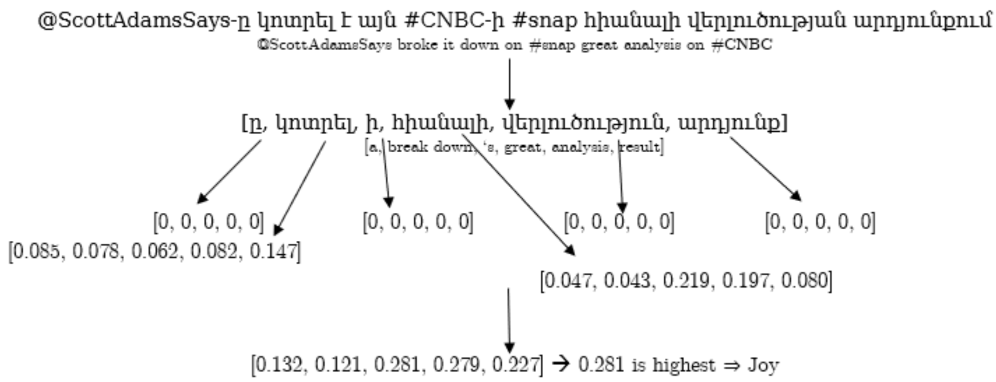

Figure 4.1: An example showing how vectorization of texts is done; according to the example, this sentence has a happy (joy) emotion

- Surprise: 6

The data set was preprocessed using the same preprocessing discussed in Section 3.3. Again, extrinsic evaluation was done using a baseline model and an ensemble-learning model.

4.3.1 *Baseline Model*

Similar to evaluating the sentiment lexicon, each word now has a vector of dimension 5 instead of 8. This is because there are inconsistencies in the number of emotions used by the lexicon and by the labeled data; not all emotions were used. Table 4.2 summarizes the mapping done between the two sets of emotions. The same mapping was used while evaluating the Arabic emotion lexicon (Badaro et al., 2018) [35].

The emotion scores falling under the same category were added together. Hence, each text now has a vector of dimension 5. Figure 4.1 shows an example. The output vector has the scores [0.132, 0.121, 0.281, 0.279, 0.227].

In this model, the emotion having the highest score determined the emotion of the text. In Figure 4.1, the highest score is 0.281, which corresponds to the emotion "joy" and hence it is labeled as "joy."

For this model, accuracy was 15.0%, recall was [17.3%, 41.9%, 11.6%, 6.4%, 50%], precision was [43.4%, 7.4%, 72.3%, 13.9%, 1.7%], and F-measure was [24.7%, 12.5%, 19.9%, 8.8%, 3.3%]. According to the F-measure, "anger" was detected best and "surprise" was detected worst.

4.3.2 *Ensemble-Learning Model*

In this second approach, GloVe (of dimension 300) is used to represent the tokens in the tokenized texts. Similar to sentiment analysis, for each lemma in the text, the global vector is retrieved. This input is fed to a support-vector machine (SVM) model. Hyperparameter tuning is done on this model. Table 4.3 shows the results of the tuning. Stratefied 10-fold cross-validation is used. For this model, accuracy

DepecheMood Emotions	Ekman Emotions
Afraid	Fear
Amused	-
Angry	Anger
Annoyed	-
Don't Care	-
Happy	Joy
Inspired	Surprise
Sad	Sadness
-	Disgust

Table 4.2: Mapping between the DepecheMood emotions and Ekman emotions

Hyperparameter	Result
C (Regularization parameter)	3792.690190732246
Tolerance	0.0001

Table 4.3: Results obtained after tuning a support-vector machine model for emotion recognition

was 59.0%, recall was [59.4%, 16.3%, 75.5%, 23.1%, 16.7%], precision was [43.9%, 38.9%, 70.9%, 43.9%, 50%], and F-measure was [50.5%, 23.0%, 73.1%, 30.3%, 25%]. According to the F-measure, "joy" was detected best and "fear" was detected worst.

On the other hand, the emotion scores of each word in a sentence are retrieved from the Armenian emotion lexicon. This input is fed to a logistic-regression (LR) model. Similar to the support-vector machine model, hyperparameter tuning was done on this model with results as shown in Table 4.4. For this model, accuracy was 52.2%, recall was [14.3%, 0%, 91.8%, 0%, 0%], precision was [32.2%, 0%, 54.5%, 0%, 0%], and F-measure was [19.8%, 0%, 68.4%, 0%, 0%]. According to the F-measure, "joy" was detected best and "fear", "sadness," and "surprise" were never detected.

Hyperparameter	Result
C (Regularization parameter)	4.281332398719396
Tolerance	5.555555555555555

Table 4.4: Results obtained after tuning a logistic-regression model for emotion recognition

The training and prediction of both models are used to make final predictions. We used the stacking classifier from Scikit-Learn[2]. Similar to sentiment analysis, the stacking classifier is trained, and predictions are made. For this model, accuracy was 63.5%, recall was [63.9%, 18.6%, 82.3%, 21.8%, 0%], precision was [50.3%, 34.8%, 74.5%, 45.9%, 0%], and F-measure was [56.3%, 24.2%, 78.2%, 29.6%, 0%].

[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html

	Baseline	SVM	LR	Ensemble Learning
Accuracy	0.149819	0.590252	0.521660	**0.635379**
Recall	0.254257	**0.381862**	0.212244	0.373244
Precision	0.277453	**0.495213**	0.173497	0.410971
F-measure	0.138642	**0.403657**	0.176465	0.376579

Table 4.5: Results obtained in emotion; the boldface numbers show the highest number in each measure

According to the F-measure, "joy" was detected best and "fear" was detected worst. Also, "surprise" was never detected. Table 4.5 shows the summary of the results. Recall, precision, and F-measure have been averaged. We notice a huge improvement over the baseline model. Also, the increase in accuracy from (SVM, LR) to the ensemble-learning model shows how the two initial classifiers helped each other to make a better prediction.

4.4 Conclusion

According to the experiments done on our Armenian emotion lexicon, the ensemble-learning model showed good results. So, the emotion lexicon can be used to make predictions to Armenian sentences with an accuracy of around 64%. In the future, we can use our extensive and higher-quality Armenian sentiment lexicon to build a better emotion lexicon. Similarly, this lexicon is also used to evaluate our BERT model in Chapter 6.

Chapter 5

Bidirectional Encoder Representation from Transformers

5.1 Introduction

Words in NLP were initially represented using word2vec models (Mikolov et al., 2013) [17]. Later, research was done in developing variations of word2vec, such as GloVe (Pennington et al., 2014) [18]. Since these word representations lacked contextualized meanings, new language models were developed. BERT (Devlin et al., 2019) [16] was one of these models. A word representation by BERT includes contextualized information that the other models lacked. Training BERT requires a huge training corpus.

For Armenian, there exist GloVe word representations in different dimensions (Ghukasyan et al., 2018) [30]. As part of expanding BERT to include other languages, Google developed multilingual BERT, which includes over 100 languages. Multilingual BERT has a good performance for many languages, but it lacks in performance when compared to monolingual BERT for non-English languages. That is why our third contribution will be to develop an Armenian BERT to enhance all Armenian NLP tasks, specifically sentiment and emotion recognition.

5.2 Methodology

We developed an Armenian language representation model to improve Armenian NLP tasks. The Armenian BERT, ArmBERT is based on the BERT model developed by Devlin et al. (2019) [16]. It is a language model, which is widely used in many NLP tasks for different languages. We used the BERT-base configuration, which includes twelve encoder blocks, 768 hidden dimensions, twelve attention heads, 512 maximum sequence length, and a total of ≈110M parameters.

5.2.1 *Pre-Training Setup*

In general, BERT is trained using two objectives. In *Masked Language Model* (MLM), some of the words are masked, and the model learns to predict them. In *Next Sentence Prediction* (NSP), the model is fed two sentences and it predicts if one sentence follows the other. For our version of BERT, we focus on the MLM task by masking 15% of the tokens. Masking is done on whole words instead of parts of a word to improve pre-training. This way, the model is forced to predict the whole word instead of getting hints from parts of the word. We have not implemented the NSP task.

5.2.2 *Pre-Training Data Set*

BERT was trained using 3.3B words extracted from the English Wikipedia and Book Corpus (Zhu et al., 2015) [40]. Training BERT requires the collection of a huge corpus. To train ArmBERT, we collected text data from Armenian news articles[1], literature[2], and publicly available Armenian text data[3]. Altogether, the collected data set consisted of 12.6 GB of data. No preprocessing was done in removing the non-Armenian words from the articles because some terms may not have been translated to Armenian and removing such words might result in information loss.

5.3 Experiments and Results

The data sets discussed in Chapters 3 and 4 were used to evaluate the BERT model. Testing is done to compare multilingual BERT (MultiBERT) and the Armenian BERT (ArmBERT). In training the BERT models, no preprocessing is done on the text data since BERT is a language model that understands text as is. In the following sections, the experiments and results on MultiBERT and ArmBERT are discussed. The tasks for evaluating these models are sentiment analysis and emotion recognition. For both tasks, the learning rate was 3×10^{-5} and the model was trained using one epoch.

5.3.1 *Multilingual BERT*

In both sentiment analysis and emotion recognition, MultiBERT was fine-tuned using the training data and tested using the testing data. For sentiment analysis, accuracy was 77.4%, recall was 97.0%, precision was 78.6%, and F-measure was 86.8%. The model had problems detecting tweets which are labeled "negative." For emotion recognition, accuracy was 68.8%, recall was [73.7%, 67.4%, 70.7%, 59.0%, 0%], precision was [56.8%, 52.7%, 88.5%, 50%, 0%], and F-measure was [64.3%, 59.2%, 78.6%, 54.1%, 0%]. The model had problems detecting tweets which are labeled "surprise."

[1] http://panarmenian.net, http://iravunk.com, etc.

[2] http://www.eanc.net/EANC/library/library.php?interface_language=en

[3] https://wortschatz.uni-leipzig.de/en/download/Armenian (Goldhahn et al., 2012) [41], http://oscar-corpus.com (Ortiz et al., 2020) [42]

The multilingual BERT was constructed using a very small corpus. This would have had negative effects on the predictive power of MultiBERT. However, from the testing done for sentiment analysis and emotion recognition, we notice that MultiBERT produced relatively good results for those tasks.

5.3.2 *Armenian BERT*

ArmBERT is fine-tuned using all the measurements, tasks, and data sets used for training MultiBERT. For sentiment analysis, accuracy was 82.5%, recall was 94.7%, precision was 84.5%, and F-measure was 89.3%. Similar to the case of MultiBERT, the model had problems detecting tweets which are labeled "negative." For emotion recognition, accuracy was 77.4%, recall was [68.4%, 65.1%, 87.1%, 69.2%, 0%], precision was [72.8%, 56%, 87.4%, 63.5%, 0%], and F-measure was [70.5%, 60.2%, 87.2%, 66.3%, 0%]. Similar to the case of MultiBERT, the model had problems detecting tweets which are labeled "surprise."

The Armenian BERT was constructed using a large corpus. This would suggest a very good performance when experimenting with ArmBERT. As the results show, ArmBERT showed exceptionally good results when performing these tasks, especially when compared to MultiBERT.

Table 5.1 shows a few examples where ArmBERT excelled in its performance when compared to MultiBERT. For sentiment, ArmBERT was able to detect the negation in the negative sentiment tweet that MultiBERT was not able to detect. For emotion recognition, MultiBERT has categorized fear, or a negative emotion, as joy, which is a positive emotion. ArmBERT had no problem identifying this as fear.

Table 5.2 summarizes the performances of MultiBert and ArmBERT in sentiment analysis and emotion recognition. For emotion recognition, recall, precision, and F-measure have been averaged.

5.4 Conclusion

The experiments performed on ArmBERT in Section 5.3.2 showed that this model produces good results for the tasks we considered. Even though ArmBERT is performing very well when analyzing sentiment or recognizing emotion, we believe that its performance will excel more when we integrate our Armenian sentiment and emotion lexicons. The next step in our experimentation includes evaluating ArmBERT in these tasks using our lexicons. Chapter 6 shows this approach.

Armenian Tweet	**English Translation**	**ArmBERT Prediction**	**MultiBERT Prediction**
Ես երազում էի, որ Ահարոն Սամուելը ուրախ չէր, որ հոկտեմբերի 3-ն էր...	I dreamed that Aaron Samuels was not happy that it was October 3...	Negative	Positive
Հուսով եմ, որ վաղ կհասնեմ Լիամի և Կլեոյի հետ:	I hope to be with Liam and Cleo tomorrow	Positive	Negative
Մի քանի բան, որոնք ինձ վախեցնում են, որոտ ասեղներ, կատուներ, սողացող տիկնիկներ, բարձրություններ և եզրեր	Some things that scare me: needles, cats, crawling dolls, heights and edges	Fear	Anger
Ատամնաբույժի մոտ գնալը երբեք ավելի հեշտ չի դառնում!! #նյարդային	Going to the dentist never gets easier!! #nervous	Fear	Joy

Table 5.1: Examples showing how ArmBERT excels over MultiBERT

	Sentiment Analysis		**Emotion Recognition**	
	MultiBERT	**AmBERT**	**MultiBERT**	**ArmBERT**
Accuracy	0.773735	**0.825199**	0.687725	**0.774368**
Recall	**0.970011**	0.946943	0.687725	**0.774368**
Precision	0.785981	**0.844650**	0.717821	**0.771354**
F-measure	0.868353	**0.892876**	0.693733	**0.772258**

Table 5.2: Results obtained in sentiment analysis and emotion recognition; the boldface numbers show the highest number in each measure for each task

Chapter 6

Integrating Sentiment and Emotion Lexicons with ArmBERT

6.1 Introduction

Deep learning models were at the heart of the success of machine learning. They excelled in performance when recognizing images and speech. Deep learning is also used in natural language processing to improve accuracy of several tasks, e.g., sentiment analysis. BERT is a deep learning model that was used in many tasks of NLP. One of these tasks is analyzing sentiment of texts and performing accurate predictions. However, deep learning models, such as BERT, suffer from poor interpretability. In other words, it is difficult to integrate sentiment knowledge into the model.

Duan et al. (2021) [43] propose a solution to this problem. They suggest a new sentiment classification model, which is based on a cascade of the BERT model and an adaptive sentiment dictionary. They call this model the Dict-BERT classification algorithm. They propose the concept of positive-negative probability ratio, which when combined with a threshold, can help BERT in making better sentiment classification. This ratio-threshold combination can be used to test how confident the BERT model is in making the prediction. If the algorithm notices a lack of confidence, it sends the input over to an adaptive sentiment dictionary, which will further evaluate the sentiment of the text. According to the authors, this approach results in a "superior performance on sentiment classification."

Inspired by their approach and the results they get, we decided to integrate the sentiment and emotion lexicons with BERT using this ensemble-learning method. Section 6.2 further discusses this approach.

6.2 Methodology

This section is divided into two parts. In the first part, an overview of the approach of Duan et al.'s Dict-BERT algorithm is discussed. In the second part, the adaptations that we did for ArmBERT and our lexicons are presented.

6.2.1 *The Dict-BERT Algorithm*

The authors in [43] used an already available BERT model for the Chinese language and a sentiment dictionary, which they adapted according to their data set, the ChnSentiCorp data set [44] (details discussed later).

BERT is initially fine-tuned using the training data. The softmax layer is used to determine the probability of the sentiment classification in different categories. A comparison between these two probabilities is made possible by the introduction of a non-linear calculation in the form of positive-negative probability ratio. This ratio is used to determine whether further analysis is needed. Using this approach, we can take advantage of both tools: BERT and the sentiment dictionary.

The positive-negative probability ratio is determined by Equation 6.1.

$$\text{Positive-negative probability ratio} = \begin{cases} \frac{P_{pos}}{P_{neg}} \text{ if } P_{pos} > P_{neg} \\ \frac{P_{neg}}{P_{pos}} \text{ if } P_{neg} > P_{pos} \end{cases} \tag{6.1}$$

where P_{pos} represents the probability of a positive sentiment and P_{neg} represents the probability of negative sentiment. It is worth noting that $P_{pos} + P_{neg} = 1$ and the positive-negative probability ratio is always greater than 1.

In a given text, if $P_{pos} > P_{neg}$, the text is classified as having a positive sentiment. The outputted probabilities, however, may vary.

- If P_{pos} is 0.9 and P_{neg} is 0.1, the text is classified as having a positive sentiment.
- If P_{pos} is 0.55 and P_{neg} is 0.45, the text is also classified as having a positive sentiment.

In both cases, the text was classified as having a positive sentiment, but there is a clear difference in the probabilities that BERT outputted. Which text was BERT more confident about in predicting the sentiment? The positive-negative probability ratio can be used to tackle this question. If we find the positive-negative probability ratio of the first text, it is 9, whereas the positive-negative probability ratio of the second text is around 1.22. We can conclude that BERT was more confident in predicting the sentiment of the first text. Therefore, the higher the positive-negative probability ratio is, the more confident the model is in making the classification. If the probability ratio is relatively low, it means that the model is struggling in correctly distinguishing the sentiment tendency of a text.

Duan et al. picked a threshold to determine whether further sentiment analysis is needed. According to them, if the positive-negative probability ratio is above this threshold, the output of the BERT classification model is directly used as the final sentiment classification. If the ratio is below the set threshold, a discrimination function of the adaptive sentiment dictionary is used to complete the sentiment classification. The selected thresholds were 1.2, 1.4, 1.6, 2.0, and 3.0.

To construct the adaptive sentiment dictionary, Duan et al. had access to a list of words with positive sentiment and a list of words with negative sentiment. A corpus may have different number of sentiment words. The higher the frequency of sentiment words is, the stronger is the emotional classification ability of the sentiment words. It is expected that the contribution of each word to the sentiment tendency is different. Duan et al. used a formula to figure out the contribution of each word in the corpus. The contribution of a word depends on the number of times it appears in the corpus according to Equation 6.2.

$$\text{Contribution} = \text{sigmoid}(\text{count}) \tag{6.2}$$

In performing text analysis, adverbs, known as degree adverbs, and negative words have great influence on sentiment tendency. The sentiment of a word and whether there are negative words or degree adverbs affect the sentiment of a text. Two scores are calculated for a text: one for positive sentiment and one for negative sentiment. If the score of the positive sentiment of a text is greater than that of the negative sentiment, it is judged that the text has a positive sentiment. However, if the opposite case is true, then the text is judged to have a negative sentiment.

To determine the score of the positive sentiment, the formula in Equation 6.3 is used.

$$\text{Positive sentiment score} = \sum_{i=1}^{N} g_i \times f_i \times c_i \tag{6.3}$$

where:

- g_i represents the contribution of adverb words. If the degree adverb appears in the context of a positive sentiment word with a window of size 4, then g_i is set to 2. Otherwise, it is set to 1.
- f_i represents the contribution of negative words. If the negative word appears in the context of a positive sentiment word with a window of size 3, then f_i is set to 1. Otherwise, it is set to -1.
- N represents the number of positive sentiment words in the text.
- c_i represents the contribution of the positive sentiment word according to Equation 6.2.

The score of the negative sentiment is determined the same way by swapping positive sentiment words with negative sentiment words.

6.2.2 *Adaptations to Dict-BERT*

Following what Duan et al. proposed in constructing a Dict-BERT model, some modifications were made to reflect our data set and different tasks.

- The threshold used is set at 14.0.

- The positive sentiment words and the negative sentiment words are the adjectives, nouns, and verbs in our Armenian sentiment lexicon. To determine if a word is negative or positive, we compare the positive sentiment score of the word with its negative sentiment score. The higher score determines the polarity of the word. The positive sentiment words are 1,480 in length. The negative sentiment words are 1,381 in length.

- The degree adverbs are the adverbs in our Armenian sentiment lexicon that have higher positive sentiment score. These are the adverbs that emphasize the sentiment words. The degree adverbs are 44 in length.

- The negative words are the adverbs in our Armenian sentiment lexicon that have higher negative sentiment score. These are the adverbs that oppose the sentiment words. The negative words are 19 in length.

- For the evaluation of f_i in Equation 6.3, negative words around a positive sentiment word would cause the text to have a more negative meaning. In this case, the positive sentiment score must decrease. So, the conditional statement in this case is modified to: "f_i is set to -1 when negative words are found around the positive sentiment word. Otherwise, it is set to 1."

- The window size set in the approach discussed in Section 6.2.1 is defined as follows:

  - We check the adverbs surrounding the sentiment words by looking at maximum four words to the left or to the right of the sentiment word.
  - We check the negative words surrounding the sentiment words by looking at maximum three words to the left or to the right of the sentiment word.

Similar approach is used to determine the emotion of a text. In this case, a list of words each showing one of the emotions: anger, fear, joy, sadness, or surprise is created using the emotion score that is the highest for each word. The same calculations are done as described previously. The list that follows highlights the differences.

- To calculate the emotion probability ratio, the emotions are divided into two groups: positive (joy, surprise) and negative (anger, fear, sadness). The highest probability is divided by the sum of the probabilities of the opposite emotion. For example, if the highest probability outputted by BERT for a text is that of joy (a positive emotion), then this probability is divided by the sum of the probabilities for negative emotions (anger, fear, sadness) to get the emotion probability ratio. The length of each list of words is summarized below.

  - Anger: 617
  - Fear: 6,540

	Positive	**Negative**
Positive	821/827	46/40
Negative	151/154	109/106

Table 6.1: Comparison between the output of ArmBERT and Dict-BERT models for sentiment analysis; the first number shows the output from ArmBERT; the second number shows the output from Dict-BERT

  - Joy: 533
  - Sadness: 479
  - Surprise: 1,002

- The threshold is set at 1.3.

6.3 Experiments and Results

6.3.1 *Sentiment Analysis*

As before, the data set used for evaluating the Dict-BERT model for sentiment analysis is SemEval-2017 Task 4 data set [37]. The length of the entire test data is 1,127. According to our threshold, only 242 tweets were sent to the dictionary for further analyses. The rest used what BERT outputs as final sentiment.

With this approach, accuracy was 82.8%, recall was 95.4%, precision was 84.3%, and F-measure was 89.5%. There is an increase in accuracy, recall, and F-measure when compared to the ArmBERT model on its own. This model was able to predict more "positive" tweets correctly, but it still struggled to detect "negative" tweets. Table 6.1 shows the difference between the outputs of both models.

6.3.2 *Emotion Recognition*

For emotion recognition, SemEval-2018 Task 1 data set [38] is used, which has a length of 554. Using the preset threshold, only 30 tweets were sent for further analyses, and the rest used what BERT outputs as final emotion.

With this approach, accuracy was 77.1%, recall was [72.9%, 67.4%, 84.0%, 69.2%, 0%], precision was [66.9%, 54.7%, 90.5%, 65.1%, 0%], and F-measure was [69.8%, 60.4%, 87.1%, 67.1%, 0%]. There is not enough noticeable change between the two models. Table 6.2 shows the difference between the outputs of both models.

Table 6.3 shows the summary of the results.

6.4 Conclusion

On its own, BERT is a very powerful deep learning model. However, it sometimes struggles to distinguish between different classes when performing supervised learning tasks. Duan et al. (2021) [43] proposed using an adaptive dictionary that helps

	Anger	**Fear**	**Joy**	**Sadness**	**Surprise**
Anger	91/97	12/13	21/15	8/8	1/0
Fear	6/8	28/29	7/4	2/2	0/0
Joy	13/23	6/7	256/247	19/17	0/0
Sadness	14/16	3/3	7/5	54/54	0/0
Surprise	1/1	1/1	2/2	2/2	0/0

Table 6.2: Comparison between the output of ArmBERT and Dict-BERT models for emotion recognition; the first number shows the output from ArmBERT; the second number shows the output from Dict-BERT

	Sentiment Analysis		**Emotion Recognition**	
	ArmBERT	**Dict-BERT**	**ArmBERT**	**Dict-BERT**
Accuracy	0.825199	**0.827861**	**0.774368**	0.770758
Recall	0.946943	**0.953863**	**0.774368**	0.587237
Precision	**0.844650**	0.843017	**0.771354**	0.554299
F-measure	0.892876	**0.895021**	**0.772258**	0.568813

Table 6.3: Results obtained in sentiment analysis and emotion recognition; the boldface numbers show the highest number in each measure for each task

BERT in making final predictions. According to them, if BERT is not very confident (determined by a probability ratio and a threshold), the text is forwarded to a dictionary, which with the help of a scoring formula, can make a final prediction.

When following their approach and implementing similar steps to our ArmBERT and our sentiment and emotion lexicons, we notice that predictions improve by 0.3%. This shows how combining different methods can increase the accuracy of the model.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this extensive project, we created two large-scale lexicons, one for sentiment and one for emotion. To further help improve the quality of Armenian NLP tasks, an Armenian BERT was also constructed.

In evaluating our lexicons and models, the following evaluations and benchmarks were used:

- For each of the lexicons, we used ensemble learning by combining the outputs of support-vector machine and logistic-regression models. The support-vector machine model outputted predictions based on the word embeddings. The logistic-regression model outputted predictions based on the sentiment or emotion score of a lemma in a text. The ensemble-learning model used these outputs to make a final prediction.

- We used the multilingual BERT and the Armenian BERT on our data sets and allowed them to determine the sentiment or emotion of the Armenian texts without interfering in their predictions.

- We used the the Armenian BERT by incorporating the corresponding lexicons. Here, we judged the predictions made by BERT by evaluating a probability ratio and comparing this ratio with a preset threshold. When we noticed that BERT was not confident in determining the sentiment or emotion of a text, we forwarded the text to an adaptive dictionary, which with the help of a scoring formula, was able to make a final prediction.

All the evaluations done on our lexicons and models show some improvements over what is already available or not available at all. It can be considered that these tools are a good addition to the Armenian NLP even though some improvements might make the contributions better suited. These improvements are discussed in Section 7.2.

The code used for evaluating the lexicons and the BERT model, the Armenian sentiment and emotion lexicons, the collected and translated data sets, and the Armenian BERT can be found at `https://github.com/nigkal/ArmenianNLP`.

7.2 Future Work

Looking back at the performance of our Armenian sentiment lexicon, we believe that we can increase its accuracy by looking for a more complete Armenian-English dictionary. This way, we can capture the sentiment scores of more words, and we may get more insights into the sentiment of a sentence.

For our Armenian emotion lexicon, we may use our expanded Armenian sentiment lexicon to retrieve the emotions of those words. We may also look for other English emotion lexicons that provide more emotions than what DepecheMood has.

The Armenian BERT may be better constructed by using more data. We may increase the sources of capturing Armenian sentences and make our BERT-training corpus even bigger. We may also integrate our expanded lexicons to see how the Armenian BERT will perform.

Since our data set was in English and translation was done by a machine, the sentences may have had flaws [45]. Our next goal is to find an Armenian sentiment and/or emotion data set that is big enough to test our lexicons and get meaningful results.

Finally, we would like to expand the Armenian NLP toolkit by introducing lexicons for Western Armenian. As this is spoken by many Armenians around the world, it would be beneficial to have tools that can analyze this form of the language as well.

Bibliography

[1] *Community standards enforcement report, third quarter 2021*, Dec. 2021. [Online]. Available: https://about.fb.com/news/2021/11/community-standards-enforcement-report-q3-2021/.

[2] N. E. Kelly and L. Keshishian, "The voicing contrast in stops and affricates in the western armenian of lebanon," in *INTERSPEECH*, 2019.

[3] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *Computing Research Repository - CORR*, vol. 271-278, pp. 271–278, Jul. 2004.

[4] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[5] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, ACM, 2005, pp. 617–624, ISBN: 1-59593-140-6. DOI: 10.1145/1099554.1099713.

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

[7] F. Bond and R. Foster, "Linking and extending an open multilingual Wordnet," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1352–1362.

[8] A. D. Software, *English & armenian dictionary*, https://download.cnet.com/English-Armenian-Dictionary/3000-2279_4-10402530.html, 2014.

[9] R. Bougie, R. Pieters, and M. Zeelenberg, "Angry customers don't come back, they get back: The experience and behavioral implications of anger and dissatisfaction in services," *Journal of the Academy of Marketing Science*, vol. 31, pp. 377–393, Sep. 2003. DOI: 10.1177/0092070303254412.

[10] S. Mohammad and T. Yang, "Tracking sentiment in mail: How genders differ on emotional axes," in *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, Portland, Oregon: Association for Computational Linguistics, Jun. 2011, pp. 70–79.

[11] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, Jan. 2008, ISSN: 1554-0669. DOI: 10.1561/1500000011.

[12] K. Knautz, T. Siebenlist, and W. Stock, "Memose: Search engine for emotions in multimedia documents," Jan. 2010, pp. 791–792. DOI: 10.1145/1835449.1835618.

[13] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, Mar. 2011, ISSN: 1877-7503. DOI: 10.1016/j.jocs.2010.12.007.

[14] J. Staiano and M. Guerini, "Depeche mood: A lexicon for emotion analysis from crowd annotated news," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 427–433.

[15] J. Howard and S. Ruder, *Universal language model fine-tuning for text classification*, 2018. arXiv: 1801.06146 [cs.CL].

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, 2013. arXiv: 1310.4546 [cs.CL].

[18] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[19] M. E. Peters, M. Neumann, M. Iyyer, *et al.*, *Deep contextualized word representations*, 2018. arXiv: 1802.05365 [cs.CL].

[20] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International Journal of Lexicography*, vol. 3, pp. 235–244, 1990.

[21] K. Denecke, "Using sentiwordnet for multilingual sentiment analysis," in *2008 IEEE 24th International Conference on Data Engineering Workshop*, 2008, pp. 507–512.

[22] B. Ohana and B. Tierney, "Sentiment classification of reviews using sentiwordnet," Oct. 2009.

[23] Y. Chen and S. Skiena, "Building sentiment lexicons for all major languages," in *ACL*, 2014.

[24] B. Liu, "Sentiment analysis and subjectivity," in Jan. 2010, pp. 627–666.

[25] C. Strapparava and A. Valitutti, "WordNet affect: An affective extension of WordNet," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal: European Language Resources Association (ELRA), May 2004.

[26] S. Mohammad and P. Turney, "Crowdsourcing a word-emotion association lexicon," *Computational Intelligence*, vol. 29, Aug. 2013. DOI: 10.1111/j.1467-8640.2012.00460.x.

[27] C. Yang, K. H.-Y. Lin, and H.-H. Chen, "Building emotion lexicon from weblog corpora," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 133–136.

[28] G. Xu, X. Meng, and H. Wang, "Build Chinese emotion lexicons using a graph-based algorithm and multiple resources," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 1209–1217.

[29] A. Abdaoui, J. Azé, S. Bringay, and P. Poncelet, "FEEL: a French Expanded Emotion Lexicon," *Language Resources and Evaluation*, vol. 51, no. 3, pp. 833–855, Sep. 2017. DOI: 10.1007/s10579-016-9364-5.

[30] T. Ghukasyan, G. Davtyan, K. Avetisyan, and I. Andrianov, "Pioner: Datasets and baselines for armenian named entity recognition," Nov. 2018, pp. 56–61. DOI: 10.1109/ISPRAS.2018.00015.

[31] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile, "Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets," Nov. 2019.

[32] W. Antoun, F. Baly, and H. Hajj, *Arabert: Transformer-based model for arabic language understanding*, 2021. arXiv: 2003.00104 [cs.CL].

[33] G. Badaro, R. Baly, H. Hajj, N. Habash, and W. El-Hajj, "A large scale Arabic sentiment lexicon for Arabic opinion mining," in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 165–173. DOI: 10.3115/v1/W14-3623. [Online]. Available: https://aclanthology.org/W14-3623.

[34] C. Fellbaum, M. al-Khalifa, W. Black, and P. Vossen, "Introducing the arabic wordnet project," 2006.

[35] G. Badaro, H. Jundi, H. Hajj, W. El-Hajj, and N. Habash, "Arsel: A large scale arabic sentiment and emotion lexicon," May 2018.

[36] G. Badaro, H. Jundi, H. Hajj, and W. El-Hajj, "EmoWordNet: Automatic expansion of emotion lexicon using English WordNet," in *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 86–93. DOI: 10.18653/v1/S18-2009. [Online]. Available: https://aclanthology.org/S18-2009.

[37] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 502–518. DOI: 10.18653/v1/S17-2088. [Online]. Available: https://aclanthology.org/S17-2088.

[38] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "SemEval-2018 task 1: Affect in tweets," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1–17. DOI: 10.18653/v1/S18-1001. [Online]. Available: https://aclanthology.org/S18-1001.

[39] P. Ekman, "An argument for basic emotions," *Cognition & Emotion*, vol. 6, pp. 169–200, 1992.

[40] Y. Zhu, R. Kiros, R. S. Zemel, *et al.*, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," *CoRR*, vol. abs/1506.06724, 2015. arXiv: 1506.06724. [Online]. Available: http://arxiv.org/abs/1506.06724.

[41] D. Goldhahn, T. Eckart, and U. Quasthoff, "Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 759–765.

[42] P. J. Ortiz Suárez, L. Romary, and B. Sagot, "A monolingual approach to contextualized word embeddings for mid-resource languages," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 1703–1714.

[43] R. Duan, Z. Huang, Y. Zhang, X. Liu, and Y. Dang, "Sentiment classification algorithm based on the cascade of BERT model and adaptive sentiment dictionary," *Wirel. Commun. Mob. Comput.*, vol. 2021, 8785413:1–8785413:8, 2021. DOI: 10.1155/2021/8785413. [Online]. Available: https://doi.org/10.1155/2021/8785413.

[44] S. Tan, *Chnsenticorp*, 2020. [Online]. Available: https://dx.doi.org/.

[45] A. Poncelas, P. Lohar, A. Way, and J. Hadley, "The impact of indirect machine translation on sentiment classification," *CoRR*, vol. abs/2008.11257, 2020. arXiv: 2008.11257. [Online]. Available: https://arxiv.org/abs/2008.11257.