Bilingual Sentiment Analysis of Spanglish Tweets

Author: Melissa Serrano

A Thesis Submitted to the Faculty of

The College of Engineering and Computer Science

In Partial Fulfillment of the Requirements for the Degree of Master of Science

Florida Atlantic University

Boca Raton, Florida

May 2017

Bilingual Sentiment Analysis of Spanglish Tweets

by Melissa Serrano

This thesis was prepared under the direction of the candidate's thesis advisor, Dr. Ravi Shankar, Department of Computer Science, and has been approved by the members of her supervisory committee. It was submitted to the faculty of the College of Engineering & Computer Science and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

SUPERVISORY COMMITTEE:

_____
Ravi Shankar, Ph.D.
Thesis Advisor

_____
Xingquan Zhu, Ph.D.

_____
Perambur Neelakanta, Ph.D.

_____
Nurgun Erdol, Ph.D.
Chair, Department of Computer & Electrical Engineering and Computer Science

_____
Mohammad Ilyas, Ph.D.
Dean, College of Engineering & Computer Science

_____ _____
Deborah L. Floyd, Eh.D. Date
Dean, Graduate College

ACKNOWLEDGEMENTS

The author wishes to express her sincere thanks to faculty and professors at Florida Atlantic University's College of Computer & Electrical Engineering and Computer Science Department. Special thanks to mis primas. #Spanglish

Abstract

Sentiment Analysis has been researched in a variety of contexts but in this thesis, we will focus on sentiment analysis in Twitter, which poses its own unique challenges such as the use of slang, abbreviations, emoticons, hashtags, and user mentions. The 140-character restriction on the length of tweets can also lead to text that is difficult even for a human to determine its sentiment. Specifically, we will analyze sentiment analysis of bilingual (U.S. English and Spanish language) Tweets. We hypothesize that Bilingual sentiment analysis is more accurate than sentiment analysis in a single language (English or Spanish) when analyzing bilingual tweets. In general, currently when we analyze sentiment in bilingual tweets their analysis is done against an English dictionary. For each of the test cases in this thesis' experiment we will use the Python NLTK sentiment package.

Bilingual Sentiment Analysis of Spanglish Tweets

List of Tables viii

List of Figures.....ix

Introduction.....1

Background and Related Work.....5

Sentiment Analysis.....9

Spanglish…….....13

Experiments……..…….....19

Overview ……20

Python NLTK Package……21

Datasets……...22

Methods……...24

Preprocessing…….……..……...24

Features……...27

Classification……...31

Evaluation……...32

Results……....34

Case 1: English 34

Case 2: Spanish ….....37

Case 3: Spanglish 41

Analysis of Overall Results ……44

Discussion and Future Work.....48

Conclusion.....50

References.....51

Tables

Table 1. Official SemEval Scores (2013, 2014, 2015) for Number One Ranked System in the Subtask Message-Level Polarity Classification ..…….7

Table 2. Example Spanglish word list ……..17

Table 3. Total tweets collected and labeled ……....…...……..23

Table 4. Training dataset ……..23

Table 5. Testing dataset ……23

Table 6. Results statistics - Case 1 ……..36

Table 7. Results statistics – Case 2 …….……..40

Table 8. Results statistics – Case 3 …….……..43

Table 9. Comparison of results statistics for all three cases ……..…….45

Figures

Figure 1: Tweet demonstrating the use of Spanglish on Twitter…….……..2

Figure 2: List of example stop words……10

Figure 3. F-score formula for classifier evaluation……11, 31

Figure 4: U.S. Hispanic Population Growth…..……..13

Figure 5: Half of 2nd Generation Latinos are Bilingual…….…15

Figure 6. Spanglish tweet example of missed sentiment..……...…18

Figure 7. Spanglish Tweet Algorithm snippet…….22

Figure 8. Regular expressions for identifying emojis……..24

Figure 9. Extract features for sentiment analysis……28

Figure 10. Raw Spanglish tweet ……29

Figure 11. raw tweet after Spanglish preprocessing……...…29

Figure 12. Extracting sentiment score from synsets……35

Figure 13. Create training set and train classifier……35

Figure 14. Confusion Matrix – Case 1……36

Figure 15. Most Informative features – Case 1…….……..36

Figure 16. Translate all words to English…….……..37

Figure 17. Confusion Matrix – Case 2…….…….……..……39

Figure 18. Most Informative features – Case 2…….……39

Figure 19. Spanglish cases uses English and Spanish Snowball stemmer……..…..41

Figure 20. Confusion Matrix – Case 3…….…….42

Figure 21. Most Informative features – Case 3……..……43

Figure 22. Example of positive emoticon introduced in 2015…….. …..……44

1.1 - Introduction

In this thesis, we will examine the accuracy of the current method used for classifying sentiment in Spanglish[1] tweets[2]. We hypothesize that by using a sentiment analysis system optimized for the analysis of Spanglish tweets, we will achieve a more accurate sentiment classification than if Spanglish tweets were preprocessed and classified using the same process as English tweets.

Sentiment in Tweets can be invaluable to companies, politicians, and other organizations since it provides a medium for a large number of users to express their raw opinions and statements. Tweets can be used to gauge sentiment of stocks, products, movies, events, political representatives and much more. Twitter gives the world their own platform to speak their mind. Since tweets are limited to 140-characters most users use slang, abbreviations, emoticons, hashtags, and user mentions, which adds a challenge when compared to sentiment analysis in traditional text.

According to a study conducted by Pew Research, a Statistical Portrait of Hispanics in the United States [1], the Hispanic community makes up 17.3% of the United States population in 2014. Recently more than ever large corporations are targeting the Hispanic-American community for sales. Companies like Toyota, Target, and Taco Bell among others are targeting bilingual speaking Hispanic-Americans in their

[1] The combination of English and Spanish commonly used amongst Hispanic-Americans
[2] A message posted on Twitter limited to 140-characters

ads. While most companies have a full Spanish language commercial on Spanish speaking television channels like Univision and Telemundo, not many have crossed into the bilingual realm, yet. Some companies have analyzed the data of this market share in the United States, and chose to target Americans that speak English and Spanish. According to an article published by MotionPoint, Companies Engaging Hispanics Win Big in the U.S. and Beyond [2], in the United States Hispanics now represent a buying power of $1.5 trillion. For Hispanic Americans both languages are part of their cultures. At home their family speaks Spanish, while other social interactions require English, so it is very common to speak both languages interchangeably. For most Hispanic Americans the languages have become so intertwined that words have emerged from mixing words and suffixes from both languages. Sometimes particular words may be used more in one language than in another, so when a person is speaking in one language but never uses that exact translation for the word in another language, they will just use both languages in one sentence as in the tweet below.

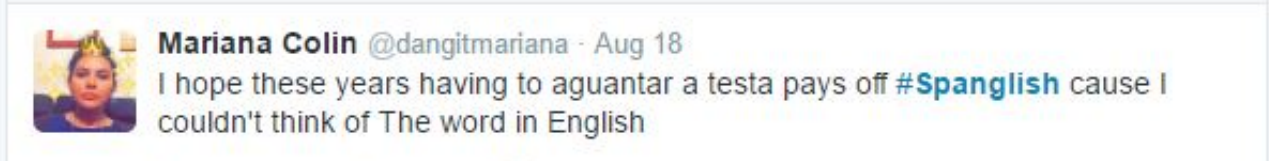


Figure 1. Tweet demonstrating the use of Spanglish on Twitter.

When Hispanic Americans see a general audience commercial for a company that is using both languages in their commercials, it gives them a feeling of "this company

gets our culture" and is therefore more likely to further look at the products the company offers. As other companies follow suit, there will naturally be a greater demand for accurate analysis of data representing bilingual Hispanic-Americans. While extensive research has been done in regards to sentiment analysis in a single language, especially in English, no research has been done to look at the accuracy of classification of bilingual tweets. They are most likely being lumped with the English tweets, being preprocessed and analyzed against a purely English dictionary. The misclassification of these tweets could mean a loss in revenue or a missed opportunity for companies looking to target the Hispanic-American community.

In 2008, Pang and Lee published a survey paper, Opinion Mining and Sentiment Analysis [3], on the topic of opinion mining and sentiment analysis where they discuss techniques and approaches to directly enable opinion-oriented information-seeking systems. Although the applications and datasets discussed are related to reviews, recommendations, business and government intelligence, and support of politicians or other public figures and these types of texts are more formally written, many researches adopted the techniques discussed in the survey paper to tweets. In the years since the launch of Twitter the interest and research in regards to sentiment in tweets grew.

SemEval[3] is an ongoing series of evaluations of computational semantic analysis systems SemEval evolved in 2007 from its predecessor SensEval[4]. SemEval is a conference which is attended by industry professors, students, and professionals where teams design tasks for which they want a problem solved related to sentiment analysis. It

[3] http://alt.qcri.org/semeval2016/

[4] http://www.senseval.org/

is the largest congregation of industry professionals interested in sentiment/semantic analysis and usually lays the foundation for solving more granular tasks within the realm of sentiment analysis. Since 2010 SemEval has included sentiment analysis as one of its tasks and since 2013 has included tasks which specifically target sentiment analysis in Twitter. The tasks for sentiment analysis in Twitter have been only for English tweets. New tasks have recently been added for textual similarity and cross-lingual analysis of general text in other languages such as Spanish or Arabic. However, they have yet to introduce a task for sentiment analysis in bilingual tweets.

Throughout the course of this thesis we will present other research done in sentiment analysis and specifically research on sentiment analysis in tweets. We will then discuss sentiment analysis strategies and techniques, and present further details regarding Spanglish as a language in sentiment analysis. Then we will present our datasets, dictionaries, and sentiment analysis packages we use to execute our three experiments. Our experiments will provide statistical analysis which will allow us to conclude and discuss the results relative to our hypothesis that Spanglish tweets will be more accurately classified when preprocessing and resources specific to the Spanglish language are applied.

1.2 – Background and Related Work

Over the past several years short informally written texts have become increasingly popular, whether they are tweets, SMS, or other microblogging platforms.

In 2008, just a couple of years after Twitter launched, Pang and Lee published a survey paper [3] on the topic of opinion mining and sentiment analysis where they discuss techniques and approaches to directly enable opinion-oriented information-seeking systems. Although the applications and datasets discussed are related to reviews, recommendations, business and government intelligence, and support of politicians or other public figures, these reviews and writings tend to be more formally written than the type of text I discuss in this paper. Nevertheless, Pang and Lee's paper has been infamously referenced through research in the field of sentiment analysis. They take us through the main challenges also addressed (although in a relative manner to informally written short text) in papers written at a later date such as: (1) extracting documents pertaining to a relevant topic, (2) extracting or identifying overall sentiment of the document, and (3) classifying and presenting the polarity of the sentiment identified in the document. They also address key concepts in classification and extraction which are also relevant in informally written short text such as classifying and extracting documents based on sentiment polarity while addressing negation, parts-of-speech, subjectivity, and topic relevance while building feature vectors or other representations.

More recently, there have been papers written which have analyzed sentiment analysis in Twitter as well as other informally written short text such as SMS and blog sentences. In the 2012 paper, *If you are happy and you know it…tweet* [4], the authors presented a cascaded classifier general framework for per-tweet processing (in contrast with batches of tweets which had been the usual form of processing the tweets). In 2009, the authors of *Twitter sentiment classification using distant supervision* [5] specifically used tweets with emoticons and hashtags expressing sentiment for their training set and then used a test set that did not necessarily contain emoticons.

As the interest in sentiment analysis research grew, along with the number of active Twitter users, sentiment analysis in Twitter became a prime interest for data scientist and researchers which resulted in its inclusion in the 2013 SemEval workshop.

We will highlight in Table 3 the number one ranking systems for the most popular subtask, Message-Level Polarity Classification, since Sentiment Analysis in Twitter became a task in SemEval in 2013 in the table below. These systems are ensemble systems which have been built of multiple proven successful analysis systems. In our analysis, we will see how the F-Score of our three experiments compare to the F-Scores obtained by these top SemEval systems, this will give us some insight on how accurate our classification is in the three different languages (English, Spanish, and Spanglish) as is relative to top Twitter sentiment analysis systems.

Year	Name of System	F1-Score (macro-average)
2013	NRC-Canada	69.02
2014	TeamX	70.96
2015	Webis	64.84

Table 1: Official SemEval scores for number one ranked system in the Subtask Message-Level Polarity Classification for the past three years.

Outside of SemEval there has also been lots of development to advance sentiment analysis in Twitter. Companies like BrandWatch [6] sell software as a service which allows you to track sentiment of a product on twitter. Semantria [7] takes this service a step further by allowing its users to analyze not only tweets, but Facebook posts, surveys, or reviews. There are also publicly available Application Programming Interfaces (APIs) that researcher, developers, and students can use to analyze sentiment such as the Python Natural Language Toolkit (NLTK)[5]. We found that the Python's NLTK is the most widely used platform for building Python programs to work with human language data. We will discuss NLTK further since this is one of the Python packages that we use for our experiments. Another useful API related to sentiment analysis in Twitter is Sentiment 140[6], it was created by Stanford students that provided us with great insight into useful features to use and appropriate pre-processing.

Most note able for us, of all the research discussed here related to sentiment analysis, is that mostly all of it has been done on English text. While no research has been done for mixed language or bilingual tweets, there has been some research done in

[5] http://www.nltk.org

[6] http://www.sentiment140.com/

the area of cross-linguistic sentiment analysis (from English to Spanish) in general text. In Cross-Linguistic Sentiment Analysis: from English to Spanish, Brooke and group [8]. The most common method for applying sentiment analysis to the Spanish language (or any language other than English) is to use Google or Bing Translate to translate a corpus from Spanish to English. There has also been research in sentiment analysis of Spanish tweets. Concuera and group [9] participated in TASS2015[7], they applied SVM machine learning model that was trained with features extracted from the TASS dataset. They extracted features typical to systems analyzing tweets such as hashtags, emoticons, elongated words, among others, and were able to obtain an F-score of 60.6.

In 2012, TASS[8] began to hold an annual conference, and in 2015 was held as part of the 31st SEPLN (Spanish Society for Natural Language Processing) Conference. TASS offers resources for research and development such as Spanish corpora related to different topics. Organizations that wish to participate in TASS compete for the system with the most accurate classification and submit research papers explaining their approach to sentiment analysis in the Spanish language[9]. A group out of the University of Texas [10] developed an algorithm for mapping SentiWordNet to the Spanish WordNet[10] in order to create their own Spanish Full Strength Sentiment Lexicon. When they used the SVM classifier they obtained a precision between 64% and 74%.

[7] http://www.sepln.org/workshops/tass/2015/tass2015.php
[8] TASS is an annual Spanish language sentiment analysis conference similar to our SemEval.
[9] http://www.cs.columbia.edu/~orb/papers/spanish_sentiment_tass_2015.pdf
[10] http://timm.ujaen.es/recursos/spanish-wordnet-3-0/

1.3 - Sentiment Analysis

Sentiment Analysis (or opinion mining) refers to natural language processing and text analysis to identify polarity of the user's sentiment in a piece of text.

In the past sentiment analysis primarily targeted formally written text such as product or movie reviews. However, since Twitter's launch in 2006 its number of active users has exploded, with over 500 million users in 2015. Users tweet their messages and at times expressing their feelings toward a product, person, or idea. Sentiment Analysis of this data is useful in many applications such as customer service, sales, politics, and stock analysis.

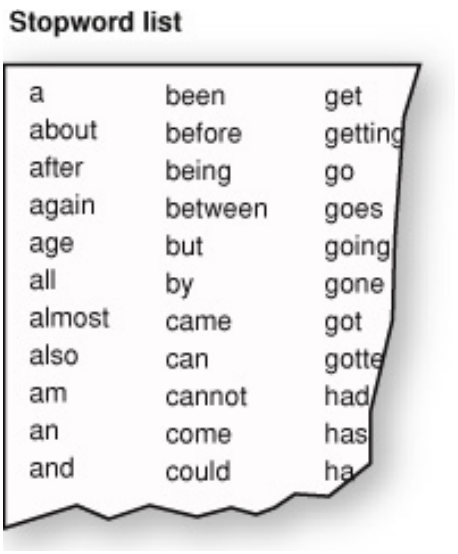
We know that Sentiment Analysis in tweets pose unique problems such as the use of slang, abbreviations, emoticons, hashtags, and user mentions. The 140-character restriction on the length of tweets can also lead to text that is difficult even for a human to determine its sentiment.

The Sentiment Analysis problem has been researched in different contexts such as product, movie, or book reviews. Sentiment analysis in Twitter differs from analysis in these contexts because of the format the text is usually written in. The text of a review is usually a bit more formal than a tweet. It will rarely have misspellings, mostly use complete sentences, and won't usually use slang or emoticons. Tweets on the other hand are very informal, written using only 140 characters, the text will use abbreviations,

slang, emoticons and other symbols, fragment sentences, accidental and purposeful misspellings.

In general, there is a process to implementing a sentiment analysis system, begin with preprocessing. Before sentiment analysis is done, a number of preprocessing steps are applied to the dataset. Common preprocessing consists of tokenizing the text, removing stop words, stemming[11] and other word normalization, and part of speech tagging.

Figure 2. Example of common stop words.



Since we will be focusing on tweets, additional preprocessing may need to be applied such as the normalization of slang or purposefully misspelled words. After preprocessing a bag-of-words model is usually created from the cleansed text. This is done by comparing each individual word to sentiment lexicons, such as SentiWordNet, labeling the tweet with a sentiment tag, and building a feature set for each tweet in the dataset. The model is then used to train the classifier which will be used as a basis in the sentiment classification of the dataset. Sentiment analysis systems that have been constructed to opinion mine English tweets over the past few years have taken different approaches to obtain a better F-Score by adding additional features sets such as: 1- to 4- grams, number of all-capitalized words, occurrence of part-of-speech tags, number of non-single punctuation marks and if last one is! (i.e. - !!, ?!,

[11] The process of removing suffixes and prefixes to only analyze the root words.

!?), emoticons, number of words lengthened by repeating a letter more than twice, and the number of negated segments. When analyzing sentiment in more formal text many of those features would not be needed, but not including them in the analysis of informally written text can cause a greater misclassification and lower F-Score. When choosing features for analyzing sentiment it is important that we do so by looking at how the text is written in the corpus which we will be analyzing.

Figure 3. The F-Score formula:

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Our analysis compares the F-score we obtain using the Python NLTK package to analyze our Spanglish tweets, with the F-scores obtained by teams participating in past SemEval Twitter tasks and TASS Twitter tasks.

A few of the classifiers most frequently used in sentiment analysis are Naïve Bayes which take the classification task from a statistical point of view, Maximum entropy, and SVM (Support Vector Machines) which applies a linear regression algorithm.

Throughout this research, we have found that the two most commonly used programming packages for sentiment analysis are Python's NLTK (Natural Language Tool Kit) and R's[12] sentiment package. Python's NLTK provides easy to use interfaces for popular lexical resources as well as an entire suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

[12] https://www.r-project.org/other-docs.html

Recently R's sentiment analysis capabilities have become extremely popular amongst data scientist because of its easy integration with data visualization tools like Tableau[13]. We will discuss Python's NLTK package in further details in the experiments section of this thesis.

[13] www.Tableau.com

1.4 – Spanglish

Since the 1990s the Hispanic population in the United States has had huge growth. According to Pew Research [1], the Hispanic-American population more than doubled from 1990 to 2010.

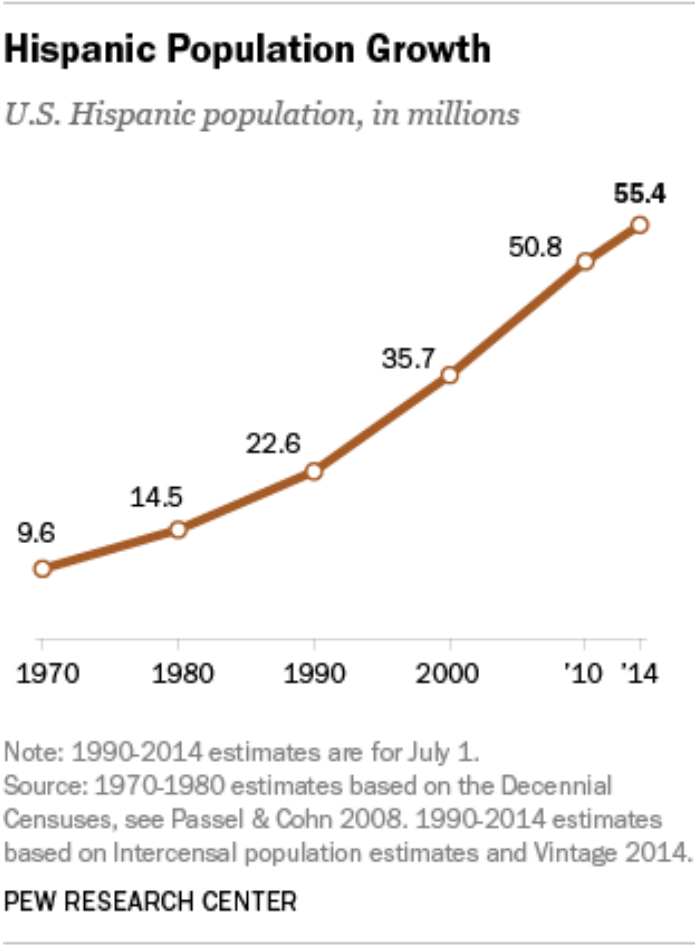


Figure 4. U.S. Hispanic Population Grow

In 2014, the U.S. Hispanic population was 55.4 million, or 17.4% of the entire United States population. The U.S. Census Bureau predicts the Hispanic population in the United States to be double what it is today by the year 2050 [11]. This is a huge market share that can be missed if not carefully targeted. Within a period of 7 days in the spring of 2015, 13.41 million U.S. viewers watched the popular Spanish television channel Univision [12]. For Hispanic Americans both English and Spanish are part of their cultures. At home family speaks Spanish, while other social interactions require English, so it is very common to speak both languages interchangeably. For most Hispanic-Americans the languages have become so intertwined that words have emerged from mixing words and suffixes from both languages. A Spanglish sentence may simply include words that are in the English dictionary as well as words that are in the Spanish dictionary; but a Spanglish sentence may have words in it that do not exist in the English nor Spanish dictionaries because the word is not a word in neither of the languages. These Spanglish words were not just invented with the emersion of Twitter or social media. Many unique Spanglish words have been spoken in Hispanic American households for generations, so even though these words do not belong to any formal dictionary they are just as much a part of the Hispanic-American communities' vocabulary as any English or Spanish word.

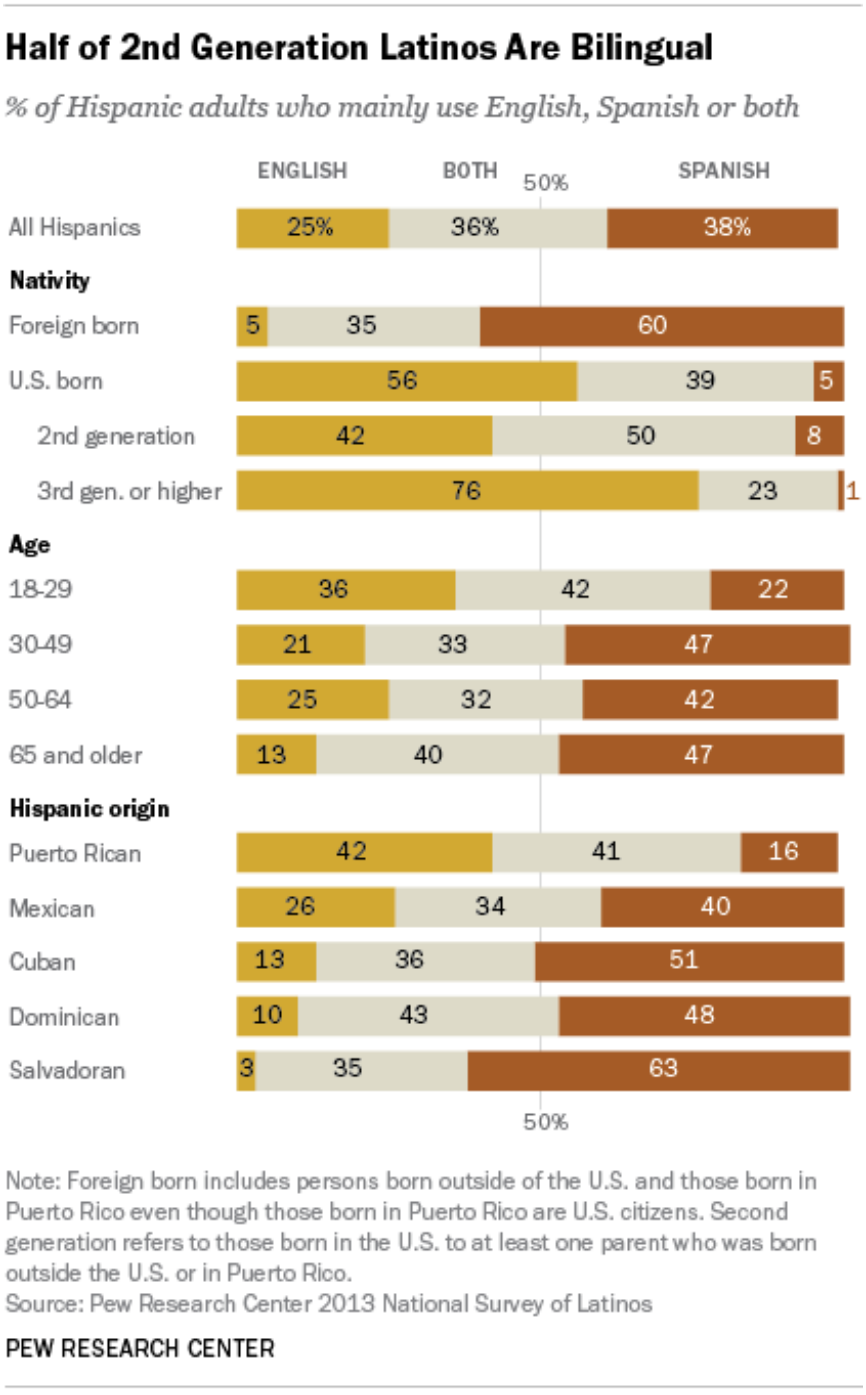


Figure 5. Half of 2nd Generation Latinos are Bilingual

With over half of second generation Hispanic Americans being bilingual, by using both languages in advertisements companies can capture the attention of the 1st and 2nd

generation of Hispanic Americans. This bilingual audience makes up for 36% of all Hispanics in the United States. Since Hispanic Americans make up for $1.5 trillion of the buying power in the United States missing the sentiment put forth from this group can be costly for any company or organization looking to use sentiment data to drive marketing.

Spanglish emerged in the Hispanic American culture by combining English words and Spanish words to create a Spanglish word which may often be more widely used than the word in either proper languages English or Spanish. Spanglish serves as a basis for self-identity and a feeling of unity in their culture and community. In this thesis, we loosely refer to Spanglish as its own language, however it is not its own language at all, it is simply a speaker who speaks in English or Spanish with a heavy influence from the other language. Often when a Spanglish speaker is talking they are switching back and forth between languages throughout the sentence: "Yo fui a la store to buy unas bananas pero when I got to la tienda ya estaban sold out." Also, Spanglish words are sometimes invented on-the-fly or have been commonly constructed by changing the suffix from English to Spanish (or vice-versa) such as appending –ear to English verbs like in lunchear, googlear, and textear. Since the predicate, which usually contains a verb, of the sentence is often indicative of expression of sentiment we hypothesize that addressing this use of Spanglish in creating our Spanglish sentiment analysis system will increase the accuracy of classification [13].

ENGLISH	SPANISH	SPANGLISH
Truck	Camión	Troca
Mop	Trapear	Mopear
Check	Verificar	Chequear/Checke
Rent	Alquilar	Rentar/Renta
Bills	Cuenta	Billes
Push	Empujar	Pucha/Pusha
Lunch	Almuerzo	Lunche
Pork	Cerdo	Puerco
Fork	Tenedor	Trinche
Supermarket	Supermercado	Marketa
Watch out	Cuidado	Watchale
Happy	Feliz	Chido
Embarrass	Avergonzar	Pena

Table 2. Example Spanglish word list

In the case of Table 2 all the Spanglish words that are listed are not words in the English nor in the Spanish dictionary.

Over the course of our data collection and experiments we found it to be fairly consistent that when we applied our Spanglish Tweet Algorithm our resulting dataset is approximately 5% of the tweets we streamed from the United States and Mexico.

Since these tweets consist of words from both the English and Spanish language, and may include words that are not found in either dictionary, using the current sentiment classification methods 5% of tweets could actually be misclassified. There may be words in a tweet expressing sentiment which are not actually part of the lexical dictionary that the words are being classified with.

Such as:

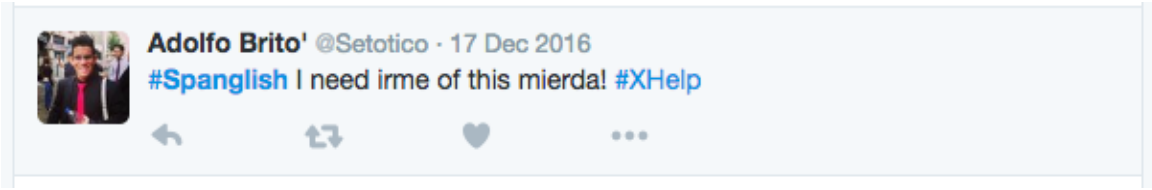


Figure 6. Spanglish tweet example of missed sentiment.

Immediately we notice words in English, however none of those words are expressing the sentiment of the tweet. This tweet should actually have a negative classification, but this would only be known if there was a specific algorithm applied, that is capable of classifying sentiment in Spanglish.

As we move on to our experiments we look to conclude that Spanglish texts will have more accurate sentiment classification if they were preprocessed using techniques to address the Spanglish language used in our corpus.

2.0 – Experiments

Our data set consists of bilingual tweets that will be analyzed consisting of the following cases using Python's NLTK package:

Case 1: Language: English

Case 2: Language: Spanish

Case 3: Language: Spanglish

We will measure each cases F-score, accuracy of classification, we will also look at the confusion matrix for each case and discuss the specificity as well as sensitivity. We expect to see Case 3 have a higher rate of accuracy of sentiment classification. In section 2.4 we will expand on the reasoning for the results of each experiment case.

2.1 – Overview

In this thesis, we hypothesize that if we build a Spanglish language Sentiment Analysis system we will obtain a more accurate classification, as oppose to the traditional sentiment analysis of these types of tweets where the corpora, lexical dictionary, and other natural language processing resources used are applied only in a single language (likely English). We will achieve this by giving attention to the way Spanglish words in tweets may be constructed such as recognizing that these tweets consist of both languages English and Spanish as well as English words containing Spanish suffixes. Concluding that when analyzing sentiment targeted at the Spanglish speaking community, certain techniques should be applied to obtain optimal classification.

In the following sections of this thesis we will discuss the details of the steps we took to conduct a blinded analysis with our experiments such as the development packages used, datasets obtained, preprocessing, features used, classification, and evaluation methods.

Our system uses the same sentiment analysis code for building and extracting feature sets for all three cases. For each of the different systems (English, Spanish, and Spanglish) the preprocessing methods were slightly different. We will discuss the differences when we get to the experiment details for each system.

2.2 - Python NLTK Package

The Natural Language Toolkit (NLTK), is a suite of libraries and programs for the Python programming language used for symbolic and statistical natural language processing. In our research, we found that NLTK is the most commonly used library in programming for natural language processing. Since sentiment analysis has gained popularity among researchers and data scientists, so has NLTK because it has most of the functionality built-in that is required to construct a simple sentiment analysis system such as classification, tokenization, stemming, tagging, and more. Since the NLTK package is so widely used many of its modules have been developed further and now offer the same functionality for a limited number of non-English languages. NLTK also provides over 50 corpora and lexical resources. In the preprocessing of our data we used the NLTK stop words corpus for English along with the English Snowball Stemmer provided by NLTK. We use NLTK SentiWordNet[14] and the NLTK POS-Tagger in building our feature set. Then we apply our features function to build our training set, and use the NLTK Naïve Bayes classifier.

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

2.3 – Datasets

To acquire our dataset of Spanglish tweets we started by using the Twitter streaming API with the python module tweepy. We streamed and collected tweets posted from within a GeoBox with boundaries outlining the United States and Mexico. Then we filter the dataset through our Spanglish Tweet algorithm so that our resulting dataset will consist of purely Spanglish tweets. Our Spanglish Tweet algorithm checks that the tweet contains a word from the English dictionary and a word from the Spanish dictionary, disregarding any word that appears in both dictionaries, while taking into account words used in hashtags and mentions.

```
if not (english_dictionary.check(word) and spanish_dictionary.check(word)):
        if english_dictionary.check(word):
                english_word = True
                #spanglish_tweets_file.write("English Word!  " + word + '\n')
        if spanish_dictionary.check(word):
                spanish_word = True
                #spanglish_tweets_file.write("Spanish Word!  " + word + '\n')
```

Figure 7. Spanglish Tweet Algorithm snippet

We separate the list of Spanglish tweets, using the first 70% for training and the last 30% for testing. The Spanglish tweets which will be used for training were manually tagged for sentiment by three Spanglish speakers, the tweet receives a vote of two out of three to be labeled with that sentiment polarity classification. These tweets were

tagged positive, negative, or neutral with the neutral tweets being thrown out to avoid a large imbalance.

Dataset

	Total	**Positive**	**Negative**	**Neutral**
Number of Spanglish Tweets	1400	567	263	570

Table 3. Total tweets collected and labeled

We streamed a total of 267,420 tweets, resulting in 13,811 Spanglish tweets. So 5.16% percent of the total tweets we streamed were classified as Spanglish according to our Spanglish Tweet Algorithm. Even when we had a few trial runs of stream tweets and running them through our Spanglish Tweet algorithm we found that the percentage of Spanglish tweets was consistently around 5%.

Training Set

	# of Tweets	**Positive**	**Negative**
Total	559	381	178

Table 4. Training dataset

Testing Set

	# of Tweets	**Positive**	**Negative**
Total	241	164	77

Table 5. Testing dataset

2.3 – Methods

2.3.1 Preprocessing

The most involved part of sentiment analysis is preprocessing the data, especially in the case of sentiment analysis on tweets. Since tweets are limited to 140-characters in length and are written very informally we must take care to do extra preprocessing in an attempt to have the cleanest data we possibly can.

As part of our preprocessing we first tokenize the tweet. NLTK does provide a simple tokenizer but because of the extra punctuation, hashtags, mentions, emojis [14], and URLs often used in tweets the tokenizer included with NLTK performs sub-par. We use python regular expressions to tokenize each tweet into words, emojis, urls, hashtags, and mentions. By using the regular expression, we can categorize our tokens into mentions, hashtags, emojis, and whether an emoji has positive or negative sentiment.

```
positive_emoticon_re = re.compile(u'['
    u'\U0001F601-\U0001F60D'
    u'\U0001F617-\U0001F619'
    u'\u263A'
    u'\U0001F642'
    u'\U0001F917'
    u'\U0001F61A'
    u'\U0001F48B-\U0001F49C'
    u'\u2764]+'
    , re.UNICODE)

negative_emoticon_re = re.compile(u'['
    u'\U0001F641'
    u'\U0001F616'
    u'\u2639'
    u'\U0001F61E-\U0001F622'
    u'\U0001F624'
    u'\U0001F626-\U0001F629'
    u'\U0001F62C-\U0001F62D'
    u'\U0001F630-\U0001F631'
    u'\U0001F633'
    u'\U0001F635]+'
    , re.UNICODE)
```

Figure 8. Regular expressions for identifying emojis.

To further our preprocessing, we used the NLTK Snowball Stemmer package because it offers stemming in both English and Spanish. The flexibility to use the stemmer in both languages played a key role in our Spanglish Sentiment Analysis system.

We constructed our list of stop words from the stop words corpus provided in NLTK. As we preprocess through our tokenized tweets, we exclude any words included in the NLTK English stop word corpus.

As we further this work we would like to further the preprocessing to pay closer attention to elongated words (ie – "hellooooo", "orrrrrale"), and after taking into account possible features of elongated words apply spelling normalization to those tokens. It would also be beneficial to apply the spelling normalization to slang or purposely misspelled words which are common in tweets or other informally written texts. As of the date of this writing, while there are some research papers written in this area there are no APIs available for this type of spelling normalization. Since the emergence of emoticons or emojis there has periodically been updates and releases of new emojis. We initially created our regular expressions for the classic emoticons, but we realized that most people on twitter are using newer emojis and in fact don't use the classic emoticons very often. All sentiment analysis research we looked at only used the classic emoticons for matching a positive or negative emoticon for feature building. Our current emoji list consists of all positive and negative emoticons, along with some common sentiment indicating emojis such as ❤, that have been released as of the date of this writing. However, as we further this work we would like to label each emoticon in the complete emoji list for positive, negative, and neutral sentiment; along with all other

emojis that may be indicative of sentiment such as 🙏 (praying hands), 👍 (thumbs up), or 👎 (thumbs down). The positive and negative emoticons that we are using do include positive and negative emoticons from the updated version and is an expanded list compared to what is commonly being used in sentiment analysis.

2.3.2 – Features

Using SentiWordNet we obtained a synset which provides a positive sentiment score and a negative sentiment score for each word. Using the sentiment score for each word we took the average of the score among all words in the tweet. If that average value is greater than zero it is considered a positive tweet score, and if it is less than zero it is considered a negative tweet score. All of our features are Boolean (True or False) values.

Our feature set includes the following features:

- Has positive tweet score: Whether the average sentiment score for the tweet was greater than zero
- Has negative tweet score: Whether the average sentiment score for the tweet was less than zero
- Word count greater than six: Whether total number of words in the tweet is greater than six
- Has adjective: Whether at least one of the words in the tweet that our POS-Tagger labeled as an adjective
- Has hashtag or mention: Whether the tweet contains a hashtag or mention
- Has positive emoji: Whether the tweet contains a positive emoticon
- Has negative emoji: Whether the tweet contains a negative emoticon

```
def extract features(tweet):
    features = {}

    if(tweet[1] > 0):
        features['has positive tweet score'] = True
        features['has negative tweet score'] = False
    elif(tweet[1] < 0):
        features['has negative tweet score'] = True
        features['has positive tweet score'] = False
    else:
        features['has positive tweet score'] = False
        features['has negative tweet score'] = False

    if(len(tweet[0]) > 6):
        features['word count greater than six'] = True
    else:
        features['word count greater than six'] = False

    if(tweet[8] > 0):
        features['has adjective'] = True
    else:
        features['has adjective'] = False


    features['hash hashtag or mention'] = tweet[12]
    features['has positive emoji'] = tweet[13]
    features['has negative emoji'] = tweet[14]


    return features
```

Figure 9. Extract features for sentiment analysis.

Since we are using NLTK there are interfaces available for lexical dictionaries for both English and Spanish, however the only sentiment lexicon dictionary available through NLTK is SentiWordNet which is only in English.

The first version of our system uses SentiWordNet for all three cases. Since NLTK only offers an English sentiment lexicon, it is somewhat common practice to use Bing or Google translate to convert foreign language words to English in order to leverage SentiWordNet. In the Experiments section of this thesis we will discuss how we handled this for each of the three cases.

As we come across hashtags and mentions we set the 'has_hashtag_or_mention' feature for the particular tweet to True. Then we strip them word of its respective # or @ symbol and then further tokenize these tokens by camel case.

Since we are analyzing sentiment in tweets as oppose to a more formally written text such as a movie or product review, ideally, we will also include more features tailored to our corpus. As we further this work we would like to expand the feature set to include number of elongated words, if the verb of the text was negative or positive, and the sentiment of the complete updated emoji list.

The following Spanglish tweet gets preprocessed and in our experiment case 3, we translate and apply our Spanglish sentiment analysis techniques, translation of Spanglish words and stemming words in both English and Spanish.

```
And I don't wanna think about it pero cuando llegue el tiempo, Chente tambien   positive
```

Figure 10. Raw Spanglish tweet.

```
([("don't", 0), ('wann', 0), ('think', 0.1346153846153846), ('arriv', 0), ('tim', 0), ('also', 0.0)],
```

Figure 11. Tweet words after Spanglish preprocessing and word scoring with SentiWordNet.

Feature set for the raw tweet in Figure 10.

- Has positive tweet score: True
- Has negative tweet score: False
- Word count greater than six: False
- Has adjective: False
- Has hashtag or mention: False
- Has positive emoji: False
- Has negative emoji: False

2.3.3 – Classification

We began our sentiment analysis classification with a simple Naïve Bayes classifier[15]. The great thing about leveraging the Python's NLTK package is that there has been an extensive amount of develop to it as well as other modules and interface which can be used with it, such as scikit-learn[16]. As we further this work, by importing the SKlearnClassifier we can easily extend our current system to train multiple classifiers such as Naïve Bayes, SVM, and Logistic Regression; with this we can then take a vote amongst the classifiers for the highest category (positive or negative). By using this voting method among classifiers available to us in these packages we could possibly improve our accuracy, reliability, and be able to provide a confidence score. The confidence score would allow us to see accuracy as it compares to all classifiers used, a high confidence score would be caused by a high classification accuracy across all or most or the classifiers.

[15] http://www.nltk.org/_modules/nltk/classify/naivebayes.html#NaiveBayesClassifier
[16] http://scikit-learn.org/stable/

2.3.4 – Evaluation

The systems were evaluated in terms of a macro-averaged F1 score, the accuracy method provided to us in the NLTK package. We used the macro-averaged F1-score available through Python' SciKitLearn package so that our F1-Score would be comparable to those obtained with SemEval and TASS sentiment analysis systems. By leveraging the confusion matrix that is available through Python's SciKitLearn package we can calculate sensitivity, and specificity. In section 2.4.1 we will compare all of these statistics for each of the experiments.

An F-score is commonly used for statistical analysis of binary classification. The precision and recall values were also calculated from the confusion matrix for each experiment. Then the F score was calculated using the formula in Figure 1.

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Figure 3. F-score formula for classifier evaluation

The experiments were conducting on an Ubuntu 16.04 virtual machine using Python code and for largely the Natural Language ToolKit available to us as a Python package. The code used for these experiments are available on Github[17].

[17] https://github.com/meMelster/spanglish

2.4 – Results

2.4.1 - Case 1: English Sentiment Analysis System

In all three of the experiment cases we use Python regular expressions to tokenize the Spanglish tweets. We split the tweet by words, emoticons, URLs, hashtags, mentions, and punctuation.

We then move on to preprocessing the Spanglish tweets after ignoring all tweets with a neutral label to avoid a large class imbalance. We remove all English language stop words by using the NLTK stop words corpus. We match our regex pattern to find mention and hashtag words, we store inside an array a Boolean value indicating whether a tweet contains a hashtag (ie #Spanglish) or a mention (ie. @Spanglish). We do this because want to use the hashtag or mention count as part of our feature set. Then we further tokenize the tweet by camel case of the hashtag or mention. Next, we stemmed all the words using the NLTK English Snowball Stemmer.

Now that we have tokenized and cleansed our Spanglish tweets we can create a feature set for training using the features we outline and describe in the Features section of this thesis. We tag each word of each tweet with a part of speech, indicated by an abbreviation. We use the part-of-speech tagger included in the NLTK package. After tagging the parts of speech for each word using the NLTK POS-Tagger, we have to

convert the part of speech tag since the abbreviations are not uniform between the NLTK POS-Tagger and SentiWordNet. Providing the tagged word information to SentiWordNet results in a synset (a list of synonyms and polarity scores) for each word. We then use an average of the senti-scores for each word among the entire tweet as the senti-score for that tweet.

```
for syn in synsets:
    score+=syn.pos_score()-syn.neg_score()
    scored_tweets.append(((word, score/len(synsets)), sentiment))
```

Figure 12. Extracting sentiment score from synsets.

Next, we extract the features and apply them to our training set and with this we can train our Naïve Bayes classifier.

```
#create training set
training_set = nltk.classify.apply_features(extract_features, tweets)

#train our classifier
classifier = nltk.NaiveBayesClassifier.train(training_set)
```

Figure 13. Create training set and train classifier.

Now using this classifier, we can pass it our test set of Spanglish tweets to obtain sentiment classification, and the accuracy scores for the classifications.

We use the show most informative features function that NLTK provides for all classifiers. This can provide us with some insightful information in regards to the features we are using and their classification. Leveraging Python's SciKitLearn package we are able to obtain the confusion matrix from which we extract our statistics.

Confusion Matrix

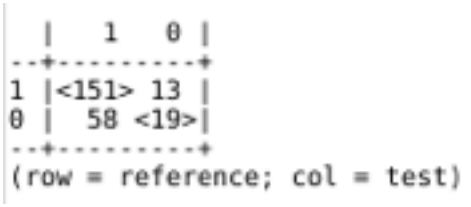


Figure 14. Confusion Matrix; 1= positive, 0 = negative – Case 1.

Most Informative Features of English Sentiment Analysis of Spanglish Tweets

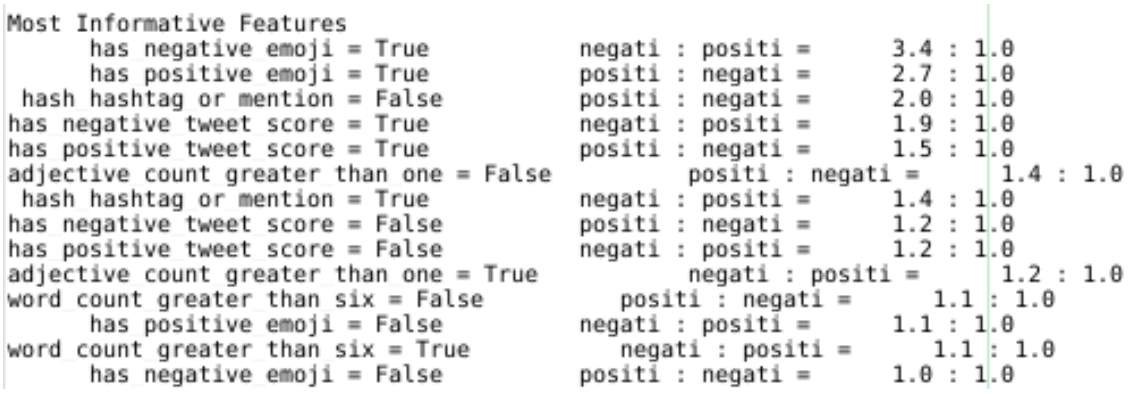


Figure 15 Most Informative features – Case 1

Results Statistics for English Sentiment Analysis of Spanglish Tweets

NLTK Accuracy	F-Measure	Sensitivity	Specificity
70.54%	57.91%	92.07%	24.68%

Table 6. Results statistics - Case 1

2.4.2 - Case 2: Spanish Sentiment Analysis System

A major challenge facing Spanish Sentiment Analysis is the lack of an equivalent to SentiWordNet in the Spanish language. Even outside of NLTK, there are limited sentiment lexical dictionaries for Spanish Language, the few that do exist do not have a very extensive dictionary. NLTK provides lexical dictionaries in other languages through WordNet, and also MultiWordNet. SentiWordNet was built by extending synsets in WordNet to include positive and negative polarity scores. This sentiment polarity label has only been extended to English language synsets.

One approach that we found to be taken most often is to use mstranslator, a Bing Translator API for python, to translate the tweets to English, so that SentiWordNet could be used. In the first version of our system we take this approach as well. When there are Spanish words with strong sentiment indications, their sentiment scores from their synsets would not be accounted for at all when using the un-translated tweets against the purely English SentiWordNet.

```
eng_words = []
for word in tweet_words:
    eng_words.append(translator.translate(word, lang_from='es', lang_to='en'))
```

Figure 16. Translate all words to English.

We chose to base our approach along these lines to keep our results consistent and comparable with the majority of classification methods currently used for these types of tweets. This will also make it comparable to our English and Spanglish Sentiment Analysis Systems.

A future experiment may be to expand the Spanish Sentiment Analysis System by using exclusively Spanish resources, in order to show the contrast of these results to the NLTK combined with a translator approach which is what is primarily used to classify sentiment in non-English tweets. We would only use Spanish stop words, Spanish stemming, and Spanish Sentiment Lexicons such as iSOL or The Spanish Sentiment Lexicon. The Spanish Sentiment Lexicon has Spanish language words that are labeled for positive and negative sentiment. They also have an offset score associated with them, however the offset score is in reference to WordNet 1.6 and NLTK is now using WordNet 3.0 to drive SentiWordNet 3.0 so the format of word offsets and scores are different. We did not take this approach because a significant amount of manual work would need to be done to either manually create a Spanish SentiWordNet 3.0 or map the Spanish WordNet 1.6 offset to the WordNet 3.0 score and then use this to drive a Spanish SentiWordNet 3.0.

Although this is our Spanish Sentiment Analysis system, after we translate words to English we are able to use the same English Snowball Stemmer and English stop words. A notable observation we made was that even though we are translating every word to English, if there is no mapping between languages for a particular word the word is still appended to the list with no change.

Now we can build and extract features for our training set in the same manner and using the same feature set as we did in the English Sentiment Analysis System in Case 1. With this training set we can train our classifier and run the test set.

Now using this classifier, we can pass it our test set of Spanglish tweets to obtain sentiment classification, and the accuracy scores for the classifications.

We use the show most informative features function that NLTK provides for all classifiers. This can provide us with some insightful information in regards to the features we are using and their classification.

Confusion Matrix

```
  |   1   0 |
--+---------+
1 |<147> 17 |
0 |  50 <27>|
--+---------+
(row = reference; col = test)
```

Figure 17. Confusion Matrix; 1= positive, 0 = negative – Case 2

Most Informative Features of Spanish Sentiment Analysis of Spanglish Tweets

```
Most Informative Features
      has negative emoji = True           negati : positi =      3.4 : 1.0
      has positive emoji = True           positi : negati =      2.7 : 1.0
has negative tweet score = True           negati : positi =      2.1 : 1.0
 hash hashtag or mention = False          positi : negati =      2.0 : 1.0
has positive tweet score = True           positi : negati =      1.4 : 1.0
 hash hashtag or mention = True           negati : positi =      1.4 : 1.0
has negative tweet score = False          positi : negati =      1.4 : 1.0
has positive tweet score = False          negati : positi =      1.4 : 1.0
adjective count greater than one = False          positi : negati =      1.2 : 1.0
word count greater than six = False          positi : negati =      1.2 : 1.0
word count greater than six = True           negati : positi =      1.2 : 1.0
      has positive emoji = False          negati : positi =      1.1 : 1.0
adjective count greater than one = True           negati : positi =      1.1 : 1.0
      has negative emoji = False          positi : negati =      1.0 : 1.0
```

Figure 18. Most Informative Features – Case 2

Results Statistics for Spanish Sentiment Analysis of Spanglish Tweets

Accuracy	F-Score	Sensitivity	Specificity
72.20%	63.03%	89.63%	35.06%

Table 7. Results statistics – Case 2

2.4.3 - Case 3: Spanglish Sentiment Analysis System

When we began constructing the Spanglish Sentiment Analysis System we knew we would need to pay much closer attention to the preprocessing so that we could adapt and extract features targeting our corpus. When we took a closer look at the way many Spanglish words are constructed and the functionality of the NLTK Snowball Stemmer, and the Bing Translator, we found translating all words to English was necessary so that they can be mapped to a synset in SentiWordNet. However, if the word requesting to be translated to the English language is not found in the dictionary then it remains the same. Many Spanglish words are constructed from English verbs with the Spanish suffix appended to it. For example: "mopear", "textar", "checkar". So, in the case of the Spanglish Sentiment Analysis System after translating all words to English, and stemming the words with the English Snowball Stemmer, we also apply the Spanish Snowball Stemmer available through NLTK.

```
word_eng_stemmed = eng_stemmer.stem(w)
#catch any Spanglish words that may not have been stemmed
spanglish_stemmed_word = spn_stemmer.stem(word_eng_stemmed)
```

Figure 19. Spanglish cases uses English and Spanish Snowball stemmer

By doing this we get the accurate English root word for many Spanglish verbs. For the examples given above applying the Spanish Snowball Stemmer returns us back: "mop", "text", and "check" respectively. Verbs can be indicative of sentiment (i.e. negative

verbs: Scream, push, hit and positive verbs: love, hug, kiss).

In addition to these verbs we built a small list of Spanglish words which many be indicative of expressing sentiment, these words are mapped to their English equivalent to ensure compatible feature sets.

By taking this approach we are able to leverage the extensive sentiment lexicon dictionary, SentiWordNet for Spanglish Sentiment Analysis as well. Now we can build and extract features for our training set in the same manner and using the same feature set as we did in Case 1 and Case 2. With this training set we can train our classifier and run the test set.

Now using this classifier, we can pass it our test set of Spanglish tweets to obtain sentiment classification, and the accuracy scores for the classifications.

Confusion Matrix

```
  |   1   0 |
--+---------+
1 |<150> 14 |
0 |  51 <26>|
--+---------+
(row = reference; col = test)
```

Figure 20. Confusion Matrix; 1= positive, 0 = negative – Case 3

Most Informative Features of Spanglish Sentiment Analysis of Spanglish Tweets

```
Most Informative Features
      has negative emoji = True           negati : positi =      3.4 : 1.0
      has positive emoji = True           positi : negati =      2.7 : 1.0
 hash hashtag or mention = False          positi : negati =      2.0 : 1.0
has negative tweet score = True           negati : positi =      1.9 : 1.0
 hash hashtag or mention = True           negati : positi =      1.4 : 1.0
adjective count greater than one = False          positi : negati =      1.4 : 1.0
has negative tweet score = False          positi : negati =      1.3 : 1.0
has positive tweet score = True           positi : negati =      1.3 : 1.0
word count greater than six = False          positi : negati =      1.2 : 1.0
has positive tweet score = False          negati : positi =      1.2 : 1.0
word count greater than six = True           negati : positi =      1.1 : 1.0
adjective count greater than one = True           negati : positi =      1.1 : 1.0
      has positive emoji = False          negati : positi =      1.1 : 1.0
      has negative emoji = False          positi : negati =      1.0 : 1.0
```

Figure 21. Most Informative features – Case 3

Results Statistics for Spanglish Sentiment Analysis of Spanglish Tweets

Accuracy	F-Score	Sensitivity	Specificity
73.03%	63.32%	91.46%	33.77%

Table 8. Results statistics – Case 3

2.4.4 Analysis of Overall Results

Most Informative Features

Our top informative features in all cases were the feature indicating whether a tweet had a negative or positive emoji or neither. We used an expanded positive and negative emoticon list compared to what is typically used in sentiment analysis which contributed significantly to the accuracy of our classifier.

Figure 22. Example of positive emoticon introduced in 2015.

We initially built the system using classic positive (☺) and negative (☹) emoticons, but we noticed that on twitter newer emojis are used much more frequently than these classic emoticons. Since our corpus consists of tweets which are informally written and written in mixed languages, identifying all emojis that express sentiment can be extremely beneficial in obtaining accurate classification.

Results Statistics for Spanglish Sentiment Analysis of Spanglish Tweets

	NLTK Accuracy	F-Score	Sensitivity	Specificity
Case 1: English	70.54%	57.91%	92.07%	24.68%
Case 2: Spanish	72.20%	63.03%	89.63%	35.06%
Case 3: Spanglish	73.03%	63.32%	91.46%	33.77%

Table 9. Comparison of results statistics for all three cases

As we hypothesized, our Spanglish experiment performs the best in all categories we measured. The results show us that our approach of stemming the words with the stemmer for both languages, along with translating Spanglish words that are not in the English or Spanish dictionary, was successful and does improve the accuracy of classifying Spanglish tweets for sentiment.

We used the macro-averaged F1-Score so that it would be comparable to scores obtained by sentiment analysis systems used in SemEval and TASS conferences. Our Spanglish sentiment analysis system obtained an F-Score just below that obtained by Webis in the 2015 SemEval as we outlined in Table 1. While the F-Score obtained from our Spanglish sentiment analysis system surpassed the F-Score obtained by Concuera and group [9] at the TASS2015 where they obtained an F-Score of 60.6.

In the English and Spanish cases, when the English Snowball Stemmer receives a word with a Spanish suffix such as “ear” or “ando”, the word does not get stemmed and we are returned the original word. So by then applying the Spanish Snowball Stemmer to the word, we are able to actually stem the word regardless of if the root word is English or Spanish.

When using the resources available to us by NLTK, used by most researchers, there still remains many words which are not tagged for sentiment and don't exist or don't directly translate from one language dictionary to another. Common text abbreviations or purposeful misspellings used by the Spanglish speaking community pose a challenge for standard NLTK libraries. Spanglish speakers are aware of these purposeful misspellings as well as the sarcastic or subjective tone which may be conveyed through reading Spanglish tweets and are hard to detect in a textual analysis system. Spanglish speakers are aware of slang words or words with alternate slang meanings, which may be used commonly in tweets. For example, a common Spanglish purposeful misspelling is the use of "k" in place of "que". This is recognized by Spanglish speakers but not our Spanglish Sentiment Analysis system. Also as a Spanglish speaker it is in their nature to adapt the language and use their knowledge of English and Spanish words to infer the meaning of words which may have been made up on the fly.

A notable observation we made in the Spanish and Spanglish cases, was that even though we are translating every word to English, if there is no mapping between languages for a particular word, the word is still appended to the list with no change.

As we compare misclassified tweets we observe that in the instances where the tweet was misclassified for all experiment cases (English, Spanish, and Spanglish), many words either do not translate to English or the literal translation does not have the same meaning as understood by a Spanglish speaker. When we look closer at tweets which were correctly classified in the Spanglish experiment case but misclassified in the other experiment cases (English and Spanish), we notice that our additional preprocessing in the Spanglish case made them be classified differently for one of two reasons. They

either had a word which was in our list of Spanglish words which are not in the English or Spanish dictionaries, and we replaced the word with its equivalent English word; otherwise the more accurate classification was attributed by applying our Spanish stemmer to the word.

We observed that in the instances of misclassification for the Spanglish case, this was usually caused by erroneous Spanish stemming on entirely English words. As we further improve on this work we would do further preprocessing to better identify Spanglish words that are constructed from English root words and Spanish suffixes, so that we would apply the Spanish stemmer only to these words.

3.1 – Discussion and Future Work

An experiment to further explore is to use lang-detect (a Python library to detect language of a text) to decide what language the tweet is (primarily) in. Then branch from that to analyze the tweet against a Spanish or English lexicon. It will be interesting to see how these results compare to our technique for handling Spanglish tweets. However this approach would only be valuable if a sentiment lexicon can be created for each language. Such as extending the synsets in MultiWordNet to include sentiment polarity creating SentiMultiWordNet, mimicking the way SentiWordNet was created by extending the WordNet synsets.

Since SentiWordNet is currently only available in English, it would be interesting to see if we applied the lang-detect library to all tweets, then if the language detected in the tweet is a language other than English we use Bing Translate (the Python mtranslate package) to convert the tweet from the given language to English. Since we know from our experiments if the word is not found in the languages dictionary it will not be translated. This should take a mixed language tweet and result in a fully English language tweet.

A top priority of future work in this area should definitely be to create a Spanish language Sentiment Lexicon, similar to SentiWordNet. There is a huge gap in sentiment classification resources in English versus what is available for other languages.

However, our most informative features lead us to a hypothesis for obtaining more accurate sentiment classification regardless of the single or multi-languages used in the tweet. Since the most informative features were if the tweet contains a positive or negative emoji, expanding this list further than what we have currently implemented would provide an implementation which would be more flexible for researchers to apply sentiment analysis to any language.

We used an expanded positive and negative emoticon list compared to what is typically used in sentiment analysis. Completing and expanding this list, it would be interesting to try a technique partially applied in *Twitter sentiment classification using distant supervision* [5] to English tweets. We could train tweets of any language, Spanish, Spanglish, Arabic, but use only tweets containing emojis for the training set. Then use tweets not containing emojis to test against. Since our currently expanded positive and negative emoji list ranks as most informative features, expanding this further to include all positive and negative emojis in the newest version of Unicode should yield these features as even more informative. If this hypothesis yields true it could be a feasible way to apply sentiment analysis to any single or multi-language tweets regardless of resources available for that language.

4.1 – Conclusion

In conclusion, this thesis provided insight into the challenges, resources available, and possible approaches to building a sentiment analysis system for bilingual tweets. While we found that about 5% of all tweets in the United States and Mexico are written using words from both the English Dictionary and Spanish Dictionary. While 5% may not seem like much, since there are millions of tweets per day, this could account for a significant gap in data that is available from this rapidly growing consumer group.

Our preprocessing was tailored to the language we were basing our sentiment analysis system on, and the features were consistent across the board. After running the experiments on the Spanglish tweets we found that the Spanglish sentiment analysis system gave us the highest accuracy, and the best F-Score. Based on these scores we conclude that when performing sentiment analysis on tweets, if the data will be used for consumer marketing and particularly if the organization wishes to understand and reach the Hispanic community, it would be beneficial for them to treat Spanglish tweets as their own language and perform specific preprocessing targeted for the language.

Spanglish is just an example of one language which has emerged from combining two languages. The study of sentiment analysis currently faces a challenge in accurately classifying text in other languages. Techniques and strategies discussed in this writing can be expanded to further the study of sentiment analysis in bilingual text, and languages other than English.

References

[1] Pew Research. "Statistical Portrait of Hispanics in the United States." Internet: http://www.pewhispanic.org/2016/04/19/statistical-portrait-of-hispanics-in-the-united-states/, April 2016 [February 2017].

[2] Chris Hutchins. "Companies Engaging Hispanics Win Big in the U.S. – and Beyond." Internet: https://www.motionpoint.com/blog/companies-engaging-hispanics-win-big-in-the-u.s.-and-beyond/, May 2015[February 2017].

[3] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1-135, 2008.

[4] Ameeta Asiaee T., Mariano Tepper, Arindam Banerjee, and Guillermo Sapiro. 2012. If you are happy and you know it… tweet. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1602-1606, New York, NY, USA. ACM.

[5] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N ProjectReport, Stanford*, pages 1-6.

[6] BrandWatch. "Understanding Sentiment Analysis." Internet: https://www.brandwatch.com/2015/01/understanding-sentiment-analysis/, 2015 [2017].

[7] Semantria LLC. Semantria Out-of-the-Box Reliability. (2015). https://semantria.com/case-studies

[8] Julian Brooke, Milan Tofiloski, and Maite Taboada. "Cross-Linguistic Sentiment Analysis: From English to Spanish." Simon Fraser University.

[9] Oscar Araque, Ignacio Corcuera, Constantino Roman, Carlos Iglesias, and J. Fernando Sanchez-Rada. "Aspect based Sentiment Analysis of Spanish Tweets." TASS 2015, September 2015, pp 29-34.

[10] Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. "Learning Sentiment Lexicons in Spanish." Internet: http://lit.eecs.umich.edu/~banea/publications/perez.lrec.2012.slides.pdf, 2012.

[11] Pew Research. "With fewer new arrivals, Census lowers Hispanic population projections." Internet: http://www.pewresearch.org/fact-tank/2014/12/16/with-fewer-new-arrivals-census-lowers-hispanic-population-projections-2/, December 2014 [February 2017].

[12] Statista. "Number of TV viewers of Univision within the last 7 days in the United States from spring 2008 to spring 2016 (in millions)." Internet: http://www.statista.com/statistics/228922/broadcast-tv-networks-univision-watched-within-the-last-7-days-usa/, 2016 [2017].

[13] Wikipedia. "Spanglish." https://en.wikipedia.org/wiki/Spanglish, 2017 [2017].

[14] Wikipedia. "Emoticons." https://en.wikipedia.org/wiki/List_of_emoticons, 2015 [2017].

[15] Boag, William, Potash Peter, Rumshisky, Anna. TwitterHawk: A feature Bucket Approach to Sentiment Analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval '2015, Denver, Colorado, June. Association for Computational Linguistics, pages 640-646.

[16] Google. "Sentiment140." (2012). http://help.sentiment140.com/home

[17] Hagen, Matthias, Potthast, Martin, Buchner, Michel, Stein, Benno. 2015. Webis: An Ensemble for Twitter Sentiment Classification. In Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval '2015, Denver, Colorado, June. Association for Computational Linguistics, pages 582-589.

[18] International Workshop on Semantic Evaluation. SemEval 2016: Semantic Evaluation Excercises. (2015). http://alt.qcri.org/semeval2016/

[19] Plotnikova, Nataliia, Kohl, Micha, Volkert, Kevin, Lerner, Andreas, Dykes, Natalie, Ermer, Heiko, Evert, Stefan. KLUEless: Polarity Classification and Association. In Proceedings of the 9th

International Workshop on Semantic Evaluation, SemEval '2015, Denver, Colorado, June. Association for Computational Linguistics, pages 619-625.

[20] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 312–320, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

[21] Preslav Nakov, Sara Rosenthal, Veselin Stoyanov, Alan Ritter. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. . In Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval '2014, Dublin, Ireland, August 2014, pages 23-24.

[22] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval '2015, Denver, Colorado, June. Association for Computational Linguistics

[23] Wikipedia. "SemEval." (2015). https://en.wikipedia.org/wiki/SemEval

[24] Wikipedia "Sentiment Analysis." (2015). https://en.wikipedia.org/wiki/Sentiment_analysis

[25] Ameeta Agrawal and Aijun An. 2014. Kea: Sentiment Analysis of Phrases Within Short Texts. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014),* August: (380-384).

[26] Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.

[27] Chris Potts. "Sentiment Lexicons." Internet: http://web.stanford.edu/class/cs424p/materials/ling287-handout-09-21-lexicons.pdf, 2010. [2017]

[28] Unicode. "Full Emoji Data, v4.0." Internet: http://unicode.org/emoji/charts/full-emoji-list.html, 2017 [2017].

[29] Apple Developers. “Stop words list.” Internet: https://developer.apple.com/library/content/documentation/UserExperience/Conceptual/SearchKitConcepts/art/stopwords_list.gif.

[30] TASS 2015. “Welcome to TASS 2015!” Internet: http://www.sepln.org/workshops/tass/2015/tass2015.php, 2015 [2017].

[31] Manasee Godsay. “The Process of Sentiment Analysis: A Study.” *International Journal of Computer Applications.* Vol. (126-No.7), pp. 0975-8887. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.6072&rep=rep1&type=pdf [2017]

[32] Stanford University. “What POS tag set does the parser use?” Internet: http://nlp.stanford.edu/software/spanish-faq.shtml#tagset [2017].