

快乐的人 / سعيد رجل
Positive, Negative, or Neutral?
A Solution for the Sentiment Analysis of
Chinese and Arabic

by

Øystein Brådland, Mats Oseland, XUE Feng, Ines Abdennadher, Brian Gonzalez

Supervisor: G-money...(UiA)

Faculty of Engineering and Science
University of Agder

Grimstad, 7 December 2010

Keywords: Semantic Orientation, Sentiment Analysis, Stanford Parser, Google Translate, Search Engine, Chinese, Arabic, PMI-IR

Abstract: Thanks to the Internet and the explosion of platforms such as blogs, forums and various other types of communications, consumers have at their disposal an arena enabling them to share their experiences and express their opinions (positively or negatively) on any products or services. The field of research our work lies in is the field of subjectivity analysis.

The aim of our work is to create a sentiment analyzer using state-of-the-art technologies to classify Arabic and Chinese texts as positive, negative or neutral. First, we take Arabic and Chinese text and translate it to English using Google Translate. This text is pre-tagged by the Stanford parser in order to obtain a set of *phrases*, or noun/adjective pairs. Next, we determine the polarity of each phrase (positive, negative or neutral) and we use Google, AltaVista, Bing, and Twitter to collect hits, which serve as the IR in the PMI-IR algorithm. We found the technologies performed differently, considering accuracy and efficiency. As for Twitter, we found that Twitter is not suitable for our sentiment analyzer.

In short, the reader will see that the technologies used in this report are viable options in the construction of sentiment analyzers. FIX FIX

PREFACE

This work is part of our degree in Master of Science program in Information and Communication Technology (ICT) at The University of Agder (UiA), Faculty of Engineering and Science in Grimstad, Norway.

This project has been carried out from September to December 2010. Previous research done by our team has led us to seek a solution for the sentiment analysis of Chinese and Arabic texts. Four of the five primary languages spoken in this research group are different, two of those being Chinese and Arabic.

We would like to send our gratitude and thanks to our supervisor Ole-Christoffer Granmo for his excellent guidance and precious advices during the fulfillment of this project.

Grimstad, 8 December 2010

Øystein Brådland
Mats G. L.
Oseland
Brian Gonzalez
Ines Abdennadher
XUE Feng

Contents

1	INTRODUCTION	6
1.1	BACKGROUND	6
1.2	PROBLEM STATEMENT	6
1.3	LITERATURE REVIEW	7
1.4	PROBLEM SOLUTION	8
1.5	REPORT OUTLINE	9
2	THEORETICAL BACKGROUND	10
2.1	HOW OTHERS HAVE SOLVED THIS PROBLEM	10
2.2	STANFORD PARSER	11
2.3	ADJECTIVES, NOUNS, AND POS TAGGING	11
2.4	SEARCH ENGINES	12
2.5	NEAR OPERATOR IN DEPTH	13
2.6	POINTWISE MUTUAL INFORMATION - INFORMATION RETRIEVAL (PMI-IR) ALGORITHM	13
2.7	WHY ARABIC AND CHINESE?	14
2.8	MACHINE TRANSLATION	15
2.9	GOOGLE TRANSLATE	15
2.10	SUMMARY	16
3	SOLUTION	17
3.1	HYPOTHESIS	17
3.2	MODELING AND PREDICTING	17
3.2.1	TESTING, VALIDATION, AND CHANGES TO THE PROTOTYPE	19
3.3	EXPERIMENT (AND COLLECTION OF DATA)	20
3.4	RESULTS	22
4	DISCUSSION	24
4.1	POSSIBLE CAUSE FOR LOW ACCURACY	24
4.2	APPLICATIONS	25
4.3	POSSIBLE IMPROVEMENTS	26
5	CONCLUSION	27
6	REFERENCES	28
7	APPENDICES	30

List of Figures

1	High-level components of the sentiment analyzer	8
2	Structure of Chinese and Arabic as compared to English	14
3	Number of primary speakers: top 5 languages (2007) [15]	15
4	From Arabic or Chinese text to phrases	17
5	Search engines	18
6	Stanford Parser Accuracy	20
7	Input and output	21
8	Final implementation of the SA	21
9	SO score range	22

List of Tables

1	Advantages/disadvantages for different search engines	12
2	Table of Abbreviations	31
3	Accuracy per search engine per language	34
4	Confusion matrix and recall & precision for Arabic using Google	34
5	Confusion matrix and recall & precision for Arabic using AltaVista	34
6	Confusion matrix and recall & precision for Arabic using Bing . .	34
7	Confusion matrix and recall & precision for Chinese using Google	34
8	Confusion matrix and recall & precision for Chinese using AltaVista	35
9	Confusion matrix and recall & precision for Chinese using Bing .	35

1 INTRODUCTION

1.1 BACKGROUND

Sentiment analysis is used automatically to determine the tonality of content. It has become fashionable since the turn of the century, however, it is much more than a fad: it is drastically changing the way the public opinions are gathered and analyzed. With the advent of weblogs, wikis, social networks, and review sites, many monitoring companies of the web are positioning themselves in this niche. The task: classify the opinions that the public publishes on the web through outlets such as Facebook, Twitter, or Epinions in hopes of capturing tone (or sentiment) as either positive, negative, or neutral.

Sentiment analysis is used to automatically determine the tonality of content. The term “sentiment analysis”, introduced back in 2001 [10, 1], is used to describe the automatic analysis of text and to research predictive value of judgments. -

Sentiment analysis is the classification of texts (articles, reviews, tickets) using techniques of language processing (natural language analysis, linguistic analysis, text mining ...) to discern the "feeling" from a text.

Tong et. al. posit that sentiment analysis is a rapidly developing field because of the value of its different applications: recommendations for cars, explanation of votes in elections, consultative advice on products, spam detection, and analysis and monitoring views to improve products or market research [10].

The concept of sentiment analysis is relevant in certain issues but not always in others, but sometimes even in those other cases as well. Current sentiment analyzers (SA) (see Appendix A for list of abbreviations use in this report) have issues when classifying texts. SAs often classify texts as neutral because they evoke both positive and negative sentiment, or the author simply didn't write with much emotion. But feelings often do not follow this mathematical formula. If a human analyzes the same text, it may be obviously positive or negative.

While sentiment analysis branches from the Natural Language Processing (NLP) community and Opinion Mining from the Information Retrieval community, both are often used synonymously as each fall within the scope of Subjectivity Analysis [11]. Our approach taps into both communities, but for the sake of consistency, we'll refer to our implementation as a sentiment analyzer. Tsytsarau and Palpanas [11] propose that sentiment analysis typically relies on four tasks: (1) identification of opinions (finding subjective text e.g. the adjective poor in the phrase “poor support”), (2) feature extraction (identification of the subject being commented on e.g the noun tree in the phrase “tall tree”), (3) sentiment classification (is the opinion positive, negative, or neutral?), and (4) visualization and measurement of results.

1.2 PROBLEM STATEMENT

An SA receives some corpus text as its input and outputs a sentiment score rating the positive or negative (or somewhere in between) assertions that are

contained within the corpus text. Our aim is to answer the following: which technologies exist to build accurate and efficient sentiment analyzers for Chinese and Arabic texts?

As stated, both accuracy (also precision and recall) and efficiency will affect how each technology is judged. For example, we might reveal a component of the SA that is 5% more accurate but 90% slower and consumes 50% more memory than another option.

1.3 LITERATURE REVIEW

Sentiment analysis was introduced by Das and Chen [1] and Tung [10] in 2001 to analyze feelings as part of the economy market. Likewise, other works on sentiment analysis have been proposed by Turney [14] and Pang et al [8].

Since 2002, a large number of articles covering sentiment analysis focus on the classification of comments and their polarity (positive or negative).

As sentiment analysis relates to our project, we'll expand and adapt methods Turney [14] outlines for sentiment analysis of English including (1) phrase extraction through an English part-of-speech (PoS) tagger and (2) semantic orientation (SO) scoring of the extracted phrases using a Pointwise Mutual Information - Information Retrieval (PMI-IR) algorithm. Semantic orientation, as Turney defines it on his blog (<http://www.apperceptual.com/>) is:

"... the evaluative character of a word. Positive semantic orientation indicates praise ("honest", "intrepid") and negative semantic orientation indicates criticism ("disturbing", "superfluous"). Our knowledge of semantic orientation is largely subcognitive."

Since Turney's approach considers English and our aim is to analyze Chinese and Arabic, we'll examine PoS-tagging and phrase extraction of these two languages in their natural state as well as pre-translation to English. Furthermore, as his PMI-IR algorithm is built upon AltaVista's (AV) deprecated NEAR operator (an operator that returns results where two words are located within some distance of each other) for information retrieval, we'll explore operator-less queries (implicit AND) to Google, AV and Twitter as well as the NEAR operator from Bing. We'll go in to depth later on in this paper on how we utilized Bing's version of NEAR.

Turney [14] presented a simple unsupervised learning# algorithm to calculate the semantic orientation (SO) score of what we'll call phrases, or an adjective/noun pair. He evaluated his methods on a large corpus of English reviews. To calculate the similarity between two words via the Web, different measures were used based on the proportion of documents in which the two words are present within a window of 10 words. Turney coined his algorithm PMI-IR. In an effort to evaluate a solution for non-English languages, in our case, Arabic and Chinese, we'll make modifications and additions to Turney's method.

Because PMI-IR is an unsupervised algorithm, the corpora must first be PoS-tagged and parsed, then phrases extracted. The Stanford Parser, a probabilistic

natural language parser that works out the grammatical structure of sentences, is an option that allows interfacing with Chinese and Arabic in their natural form. In essence and for our purposes, the Stanford Parser acts like a glorified PoS tagger that will aid in the extraction of phrases. Likewise, Google Translate (GT), a statistical machine translator, also translated Chinese and Arabic to English. Parsing the corpora in its natural form as well as pre-translating the corpora to English then parsing it are two options which we'll explore in depth.

1.4 PROBLEM SOLUTION

What is exactly the problem your solving / approach Our charge is to build an SA for Chinese and Arabic using modern methods and components while analyzing accuracy and efficiency for each of these methods and components.

Most SAs attempt to find the overall sentiment of a document. However, our approach is more micro than macro, that is, we're interested in SO scores for individual phrases that exist within a text rather than the sentiment analysis of the entire text. In particular, the high-level components are shown below in figure 1.

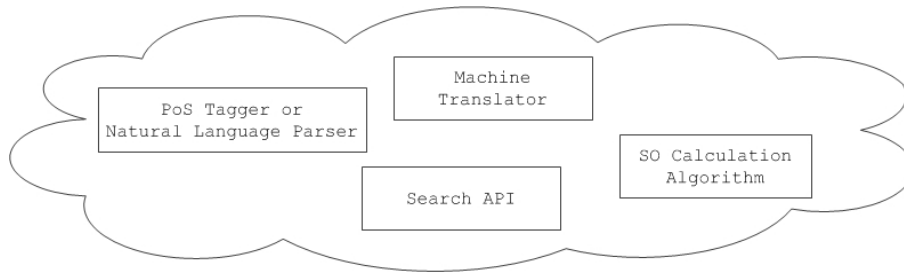


Figure 1: High-level components of the sentiment analyzer

Research Questions

Are ample amounts of the author's sentiment, tone, and intent captured during machine translation? Can we achieve accurate PoS tags of non-English languages such as Chinese and Arabic with today's technology? Most of the web is built upon English, is there adequate web content available in Chinese and Arabic to accurately implement PMI-IR? Previous research has shown that conventional search engines (Google, etc.) work well for IR; will Twitter's 140-character bits of text, known as Tweets, perform well for IR?

Our motivation

Machine learning is one of the fastest growing research fields in Computer Science. Sentiment classification by using machine learning techniques is an interesting topic. We as researchers have many motivations for this field. Four of the five primary languages spoken in this research group are different, two of those being Chinese and Arabic. Because the majority of previous work in this field has been done on English, we'd like to further expand it to these two languages. The implications of this are vast: ranging from the ability to give a scarcely spoken language a voice to allowing a business to hear this voice and expand its market.

1.5 REPORT OUTLINE

This report is organized as follows. In Chapter 2, we present previous works of other researchers, the tools and the methods that will help us to solve our problem are briefly described, and we introduce the languages of interest. We present our solution in Chapter 3. Starting with our hypothesis, we then model and predict our solution, and then obtain and visualize our results. In Chapter 4, we discuss and evaluate our work and in Chapter 5, we draw a conclusion.

2 THEORETICAL BACKGROUND

In this chapter, we'll start by talking about how other have solved this problem. Because many of the previous solutions either didn't expand on their implementation, had gaps in their implementation, or simply relied on the work of previous researchers, we'll exhaustively explore tool for the construction of an SA such as: POS taggers like the Stanford Parser, IR mechanisms such as Google, AV, Bing, and Twitter, SO calculation algorithms such as PMI-IR, and machine translators like Google Translate. Because much of our work falls under NLP, we'll give a brief overview of the languages at the core of our work, Chinese and Arabic, and break down the structure of a basic sentence in each of these languages.

2.1 HOW OTHERS HAVE SOLVED THIS PROBLEM

As stated in the Literature Review in section 1.3, most of our work is modeled off of Turney's [14] work covered in "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews". As the title suggests, Turney measured his PMI-IR algorithm with English reviews from the <http://epinions.com> website. On this site, a user can rate a product on a scale of 1 to 5 stars, 5 being the best or most recommended. Turney's approach worked as so: run a given review through the SA and get SO scores for all phrases within the review, if review was given a high rating (> 2.5 stars) by the user and the average of the SO scores was positive, then the SA correctly classified the entire review. The converse for a low rating (< 2.5 stars).

Furthermore, Zhu et al explored computing semantic orientation of Chinese words based on HowNet in 2006 [17], which is "an on-line common-sense knowledge base unveiling inter-conceptual relations and inter-attribute relations of concepts as connoting in lexicons of the Chinese and their English equivalents" [16]. Their algorithm used HowNet's inter-concept relations and inter-attributes relations to classify the unit. During their tests, they chose 40 pairs of positive and negative words as vocabulary words and 6000 words as test sets. Their algorithm achieved between 62% and 68% accuracy, considering all test sets.

Li et al researched SO in 2009 [6]. In the experiment, the authors chose the Contemporary Chinese Language Orientation Usage Dictionary as the source of vocabulary words, and extended them by synonyms dictionary. Furthermore, to disambiguate the multi-orientation words, Bigram theory was adopted. Finally, the result of the experiment was that it reached 79.31% for positive words and 78.18% for negative words.

Although they used a large number of vocabulary words obtained from two dictionaries, their method still has some disadvantages in that there are still many contemporary words and phrases generated every day that aren't included in the dictionaries. Therefore, calculating the SO of these words may lead to inaccurate classification.

2.2 STANFORD PARSER

The Stanford Parser ¹ [5] is a probabilistic parser written in Java, with several grammars (German, Chinese, Spanish and Arabic). The Stanford Parser is one of the most modern methods for PoS tagging and parsing text. Stanford’s NLP Group claims their parser to be robust and fast (because it is based on a stochastic grammar) and produces similar structures for sentences semantically equivalent in 93% of reviewed cases [5].

2.3 ADJECTIVES, NOUNS, AND POS TAGGING

As mentioned earlier, sentiment analysis relies upon finding subjective text and then identifying what exactly the subjective text is commenting on. PoS tagging is often utilized in sentiment analysis, as it deconstructs a sentence, revealing the components an author used. This is analogous to a mechanic removing parts of a car as to see the parts a manufacturer used during production, thus, to a degree, judging the manufacturer’s intent. Deconstructing a sentence helps one analyze the author’s intent.

A PoS tagger is an essential component of an SA. Run through a PoS tagger, the sentence “The early bird catches the worm.” is returned as:

The/DT early/JJ bird/NN catches/VBZ the/DT worm/NN ./.

These tags are valuable insight for a SA. With simple knowledge of tags, one can see that the author believes the bird (the subject, where NN noun) to be early (the opinion, where JJ adjective). Evidence has shown that an NN paired with a JJ, or a noun with an adjective, are good indicators of sentiment [8]. From a sentiment point of view, an adjective like dense or nouns like man or pudding are ambiguous when isolated. But paired, the phrases “dense pudding” or “dense man” become laced with sentiment, where “dense pudding” could be deemed positive or neutral while “dense man”, negative. By placing two words into a pair (in this report, we will refer to such a pairing as a phrase), the meaning becomes constrained and less attention needs to be focused on mapping words to concepts.

¹In our experiment, we have used version 1.6.1 of the Stanford Parser (current version 1.6.4)

2.4 SEARCH ENGINES

Information Retrieval, or IR, is the process of searching for documents or text within a larger set of documents or text. IR can be implemented via search engines, such as those mentioned below.

Our implementation explores four different means of IR: Google, AltaVista, Bing, and Twitter. Table 1 breaks down each of these engines.

	Advantages	Disadvantages
Google	<ul style="list-style-type: none"> • Most widely used² • Indexes over 1012 URLs³ • Results in over 100 languages 	<ul style="list-style-type: none"> • Restriction on frequency of queries • No NEAR Operator
AltaVista	<ul style="list-style-type: none"> • Widely used in SO field prior to deprecation of it's NEAR operator[14, 12] 	<ul style="list-style-type: none"> • Restrictions on frequency of queries • No NEAR Operator • Results in only 41 languages
Bing	<ul style="list-style-type: none"> • NEAR operator • No restriction on frequency of queries 	<ul style="list-style-type: none"> • Results in only 45 languages
Twitter	<ul style="list-style-type: none"> • Its users usually post snippets of text containing just the necessary words for describing a product, concept or their feelings 	<ul style="list-style-type: none"> • Returns a maximum of 1,500 hits • 140-character results

Table 1: Advantages/disadvantages for different search engines

2.5 NEAR OPERATOR IN DEPTH

The NEAR operator makes it possible to search for two words that are positioned X words from each other. Turney used AV's (when it was available) NEAR OP, which returned results where word (or phrase) X is within 10 words of word Y, with "X NEAR Y" as the query. Bing's NEAR: operator is more flexible as it allows the user to specify the distance between X and Y by appending a value to the end of the operator. An example Bing search with the NEAR operator: "Poor near:10 support" (without the quotes) would return pages where the words "poor" and "support" are separated by nine words or less.

2.6 POINTWISE MUTUAL INFORMATION - INFORMATION RETRIEVAL (PMI-IR) ALGORITHM

The PMI-IR algorithm collects statistical information through IR and then uses PMI for analysis. The core idea is that "a word is characterized by the company it keeps" [2].

The PMI-IR algorithm is a 3-step process: first, identify a word or phrase to calculate the SO score for; second, get the numbers of hits for the phrase, the phrase with one positive, and the phrase with one negative adjective; third, calculate the SO score using the values from step 2.

First, a formal definition:

$$PMI(word_1, word_2) = \log_2 \left(\frac{p(word_1 \& word_2)}{p(word_1)p(word_2)} \right)$$

The numerator of the above equation calculates the probability that two words co-exist together and the denominator calculates the probability that word1 and word2 occur independently. The logarithm of this fraction is "the amount of information that we acquire about the presence of one of the words when we observe the other" [14].

Though PMI is used above with two words, it can also be used with an adjective/noun two-word pair, called phrase – for example "red hat" and "good". Now consider the semantic orientation (SO) score calculation which takes two types of adjectives: one negative and one positive.

SO(phrase) formula

As different search engines have different search operators (e.g. Google = AND, Bing = NEAR) we use OPERATOR to denote various operators. After some algebraic manipulation, we have:

OPERATOR for Google and AltaVista: Implicit AND

OPERATOR for Bing: NEAR:25

$$SO(phrase) = \frac{\sum_{i=1}^{15} \sum_{j=1}^{15} \log_2 \left(\frac{hits(phrase \text{ OPERATOR } positive_adjective_i) hits(negative_adjective_j) + 10^{-4}}{hits(phrase \text{ OPERATOR } negative_adjective_j) hits(positive_adjective_i) + 10^{-4}} \right)}{\left(\sum_{i=1}^{15} \sum_{j=1}^{15} (i+j) \right) + 10^{-4}}$$

The SO score gives insight into the tone or feeling of the phrase. The function `hits()` returns the amount of hits for a search engine query.

According to Turney, if the SO score is positive, the phrase is positively oriented, and if the score is negative, the phrase is negatively oriented. The absolute value of the score indicates the intensity of the orientation. During Turney's tests, the PMI-IR algorithm was able to correctly classify text as positive or negative at a rate of 74.39% [14].

2.7 WHY ARABIC AND CHINESE?

Over 300 million people and 1.2 billion of Muslims use Arabic in religion (Qur'an, prayers). Despite the growth in the interest of the Arabic language⁴, the tools used to analyze the language are not robust [4]. The Arabic language differs from other Indo-European languages like English and French. It consists of 28 letters, not 26 letters: 25 consonants and 3 vowels specified for Arabic which represent long vowels. Moreover, Arabic language differs in that it has the singular and the plural like other languages, but also the double to present two persons. This difference between Arabic and Indo-European languages is syntactic, morphological and semantic. Arabic's vocabulary is rich, but very complex. Chinese is equally is not more complex. Figure 2 shows some morphological differences between Chinese, English, and Arabic.

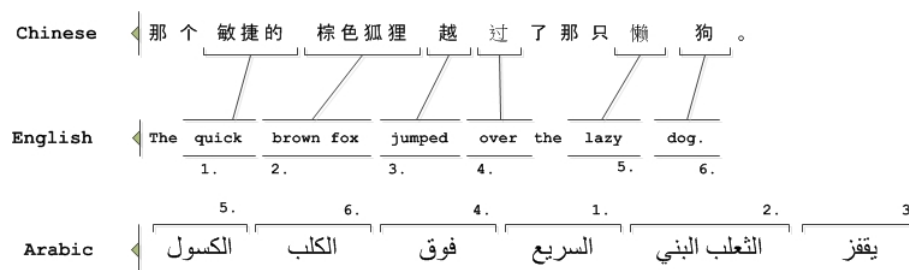


Figure 2: Structure of Chinese and Arabic as compared to English

Chinese has the most primary speakers in the world as China's population is more than 1.34 billion⁵. In many aspects, Chinese differs completely from Western languages. Unlike English, Chinese does not use the Latin (or Roman) alphabet. Furthermore, Chinese is a pictograph language in which a reader can deduce the meaning of a word through structure and style. At the same time, although English is the most widely used language, Chinese is a close second based on the largest population and the increasing number of foreign people who learn Chinese. This is shown in figure 3.

⁴Arabic Speaking Internet Users Statistics: <http://www.internetworldstats.com/stats19.htm>

⁵Number of Chinese Speakers: <http://www.cpirc.org.cn/index.asp> (in Chinese)

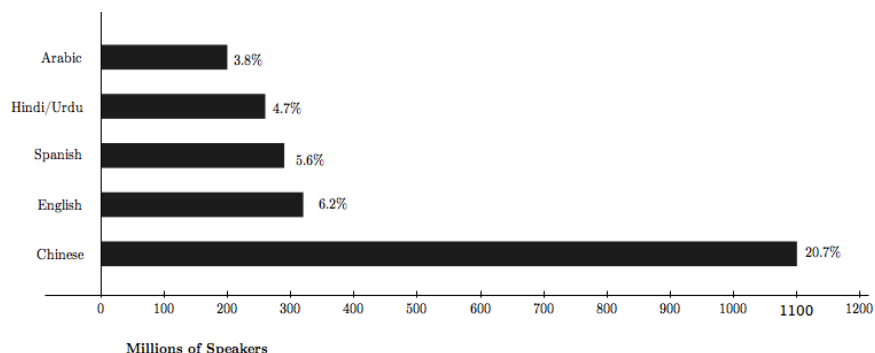


Figure 3: Number of primary speakers: top 5 languages (2007) [15]

2.8 MACHINE TRANSLATION

A machine translator is an entity that seeks to translate text or speech from one natural language to another. How can machine translation performance be measured? *The closer machine translation is to professional human translation, the better it is.* Not all machine translators are created equally.

Take for instance the following Simplified Chinese sentence:

correctly translated to English as “The early bird gets the worm.” GT translates accurately as “The early bird gets the worm” while Yahoo!’s Babel Fish (an alternative machine translator) offers “Gets up early the bird has the insect to eat.” Obviously the GT translation is more accurate.

Now take for instance the following Arabic sentence:

directly translated to English as “Merry Christmas and a Happy New Year. GT translates the phrase as “Congratulations on the occasion of Christmas and New Year” while Bing Translator offers “Most beautiful congratulations Christmas and new year.” Neither translation is completely accurate, however both capture the tone and the general meaning of the saying.

In short, machine translation is a quick and efficient method for translating languages. While the other aspect of translation, human translation, is often more accurate, it is much less efficient because a human must manually translate the entire text. A human fluent in the initial as well as the target language is a requirement for human translation. In the next section, the reader will see that GT offers translation to and from 51 different languages. We assume there are very few (if any) humans that are fluent in 51 languages.

2.9 GOOGLE TRANSLATE

Google Translate is a free statistical machine translator that seeks patterns in large amounts of text to determine which translation is best. GT is versatile as it can translate 51 languages ⁶. In spite of this fact, flaws exist in translations

⁶The official Google translate blog: <http://googletranslate.blogspot.com/>

of Chinese and Arabic to English.

A common technique used for judging machine-translation is called the BLEU (BiLingual Evaluation Understudy) algorithm. The BLEU score for a ChineseEnglish Google-translation is 32.5 and ArabicEnglish, 34.0#. The BLEU score measures the distance between a machine translation and several possible human translations for the same text . A BLEU score of 100, sometimes normalized to 1, is the result of running a BLEU test against two identical documents.

2.10 SUMMARY

In this chapter, we presented a theoretical study of tools and methods that will be used for the implementation of our SA. In the next chapter, we'll reveal the implementation of these technologies in our SA.

3 SOLUTION

Our solution, as you'll read below, is dynamic where changes were sometimes made during the experimentation process. For a complete list of the tools used in this implementation and their locations, please see Appendix 7.

3.1 HYPOTHESIS

Research has led us to state-of-the-art technologies in the field of sentiment analysis. We used many of these technologies to implement the SA. Previous research on the Stanford Parser found an F-score (sometimes called F-1 score or F-measure), a measure of both precision and recall, of $F = 83.3$ and $F = 77.4$ for Chinese [7] and Arabic [5], respectively. F-score was measured by comparing a professional human PoS-tagged text to Stanford Parser PoS-tagging of the same text. Likewise, research has shown GT to be an accurate tool for Machine Translation. Both the Stanford Parser as well as GT have solutions for working with natural Chinese/Arabic, therefore, we'll explore both of them. However, we hypothesize the former to be a better solution because it has out-of-the-box functionality to parse Chinese and Arabic naturally without the need to access another API for machine translation.

Secondly, we predict that the three search engines (Google, AV and Bing) that are currently responsible for 95% of the web's searches# will provide accurate results for the Information Retrieval part of the PMI-IR algorithm. Another option which we will explore for IR is Twitter – which returns short, emotional bits of text. We hypothesize that using state-of-the-art components and methods (those mentioned in Chapter 2 of this report) to build an SA to analyze Chinese and Arabic texts will result in accurate SO scores.

3.2 MODELING AND PREDICTING

The prototype SA (using the final model shown in figure 8 in sub section 3.3 for reference) is designed as follows: (1) input a set of Chinese/Arabic text known as the corpora T, (2) parse the text using the Stanford Parser (figure 4) to receive a set of adjective/noun pairs called phrase#, where phrase T,

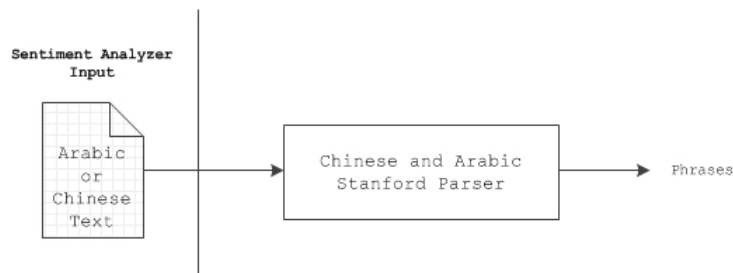


Figure 4: From Arabic or Chinese text to phrases

Figure A: (3) pair each phrase with each positive-adjective $V_{\text{ocabpositive}}$, and each negative-adjective $V_{\text{ocabnegative}}$ where $V_{\text{ocabnegative}} = V_{\text{ocabpositive}} = 15$ (e.g. “harsh life happy” or “harsh life horrible”), (4) search each of these pairs using each of the search engines (figure 5) which returns the amount of hits for each pair,



Figure 5: Search engines

and lastly, (5) calculate the SO of each phrase using the PMI-IR algorithm. The SA returns a set of SO_{phrase} values for each phrase per search engine, SG_{Google} , SAV , SB_{Bing} , where S_{phrase} is S .

In Step 4, we utilized the Search API Google, AV, and Twitter’s default search operator is AND (called ‘implicit AND’). This operator is supposedly included between each word when you search for any given combination of words i.e. “Coca X Cola” where X = implicit AND. We assumed that searching with the implicit AND would be the same as searching with the explicit AND, but it seems that “poor support” (implicit AND) and “poor AND support” (explicit AND) does not return the same amount of search results. There exists no documentation from Google or AV on this phenomenon, and for the purposes of our model, it will be ignored. Previous studies have shown the AND operator to be inferior to the NEAR operator when it is used for IR in the PMI-IR algorithm [2], but the lack of a NEAR operator leaves no other options.

Bing appears in our model because it supplies a NEAR Operator. We used “NEAR:25” (see Theoretical Background for the meaning of “:25”) during our implementation. The number 25 was arbitrarily increased from Turney’s 10 to compensate for discrepancies and flaws in translation.

Prior to Step 4, we focus on our vocabulary sets. These sets, $V_{\text{ocabnegative}}$ & $V_{\text{ocabpositive}}$, each contain 15 positive and negative English adjectives (not Chinese or Arabic adjectives for reasons we’ll see later on). For example, $V_{\text{ocabpositive}} =$ "good", "excellent", , "happy", $V_{\text{ocabnegative}} =$ "horrible", "grumpy", , "thoughtless". In step 3, we pair each element from $V_{\text{ocabnegative}}$ & $V_{\text{ocabpositive}}$ with a phrase in T (for all phrase in T).

Let’s take a look at an English example, for clarity. Consider “harsh life”, a phrase extracted from the corpora, and “good” and “horrible”, elements of

each positive and negative vocabulary set. The example Bing queries needed to calculate the SO score would then be: “harsh life”, “harsh life NEAR:25 good”, and “harsh life NEAR:25 horrible”. This is where Turney’s PMI-IR in below comes in.

$$\text{Let } r = \text{phrase}, p_j = \text{positive_adjective}_j, n_k = \text{negative_adjective}_k$$

$$SO(r|p_j, n_k) = \log_2 \left(\frac{\text{hits}(r \text{ OPERATOR } p_j) \text{ hits}(n_k) + 10^{-4}}{\text{hits}(r \text{ OPERATOR } n_k) \text{ hits}(p_j) + 10^{-4}} \right)$$

Where:

OPERATOR for Google and AltaVista: Implicit AND

OPERATOR for Bing: NEAR:25

The results from each of these queries are the values used in the PMI-IR algorithm to determine the SO score for one element in matrix M (shown below).

$$M = \begin{bmatrix} SO(r, p_1, n_1) & \dots & SO(r, p_{15}, n_1) \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ SO(r, p_1, n_{15}) & \dots & SO(r, p_{15}, n_{15}) \end{bmatrix}$$

The phrase “harsh life” is undoubtedly negative, and our comment before: a phrase “is characterized by the company it keeps”, will become evident when matrix M is averaged and the SO score is calculated. That is, pairs like “harsh life horrible” will return more hits than “harsh life good”, which will sway the SO score in the negative direction.

As mentioned in Chapter 2, the hits are the building blocks of the PMI-IR algorithm and are used to compute the SO score for each phrase. To calculate SO score for one phrase element of T, we average the elements of matrix M.

Steps 3-5 are all considered elements of PMI-IR. Steps 3-4 are the necessary steps for Step 5, the mathematical process of the PMI-IR algorithm. The equation from Step 5 calculates the SO scores for each phrase. It is important to note that this is run per phrase T.

The function hits() returns the search results for each search engine within the Search API. 10⁻⁴ is arbitrarily added to offset the denominator and avoid division-by-zero and added to the numerator to prevent SO scores of 0.0. In addition, queries that return < 4 results (chosen arbitrarily) are deemed useless and omitted.

3.2.1 TESTING, VALIDATION, AND CHANGES TO THE PROTOTYPE

Once the components of the conceptual model were constructed, we decided to test each of them individually before final implementation. Twitter’s API limits the amount of hits to 1,500. Many of the phrases extracted from our texts were quite unique, and when unique phrases are searched on Twitter, they return 0 results.

For example, parsing a sample sentence from one of the Arabic corpora returns the phrase which means “Arab citizen”. Remember that in our model each phrase must be paired with each positive-adjective $Vocab_{positive}$, and each negative-adjective $Vocab_{negative}$ where $Vocab_{negative} = Vocab_{positive} = 15$ prior to searching. A sample phrase + vocab pair: which means ‘Excellent Arab citizen’, returns 0 hits on Twitter. We found that the majority of the phrases extracted from the corpora were too unique for Twitter to return any meaningful data, and for this reason, decided to abandon the use of Twitter in our solution.

Likewise, the developing Stanford Parser claims to accurately parse Chinese and Arabic, as stated in the hypothesis. In spite of this, many actual phrases located within the corpora were not output by the Stanford Parser, which was verified by the Chinese and Arabic speakers within the group. The Chinese serializer of the Stanford Parser returned 202 phrases for one of the sample corpora written in Chinese. The same corpus, pre-translated to English using GT and then run through the English serializer of the Stanford Parser returned 501 phrases – nearly 300 more. Additionally, figure 6 shows that parsing natural Arabic using the Arabic serializer of the Stanford Parser is nearly as poor.

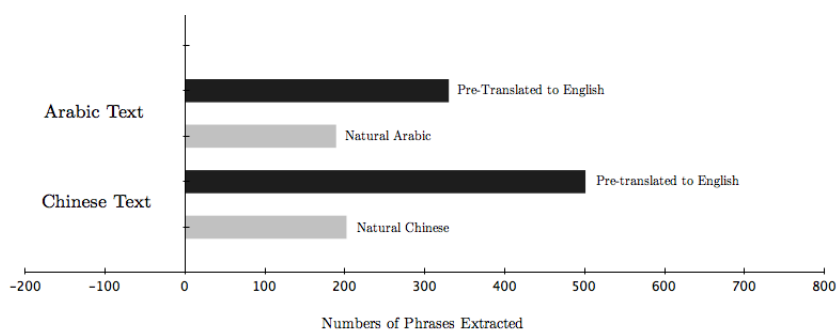


Figure 6: Stanford Parser Accuracy

3.3 EXPERIMENT (AND COLLECTION OF DATA)

We began collecting data with the alteration from the prototype shown in figure 7, taking in to account what was discussed in Section 3.2.1, “Testing, validation, and changes to the prototype”.

The figure below shows that the SA’s input is a set of Chinese and Arabic texts (the corpora). These are immediately translated by GT, giving us their English counterpart. The novelty of this alteration is unique because GT currently provides automatic language detection for 51 languages (meaning the SA can now analyze 49 languages other than Chinese and Arabic).

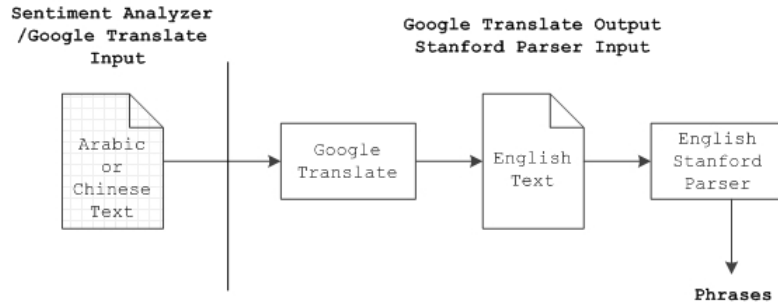


Figure 7: Input and output

Once the corpora is translated by GT, the English serializer of the Stanford Parser was used to extract phrases. The extracted phrases are composed of an adjective and a noun that the parser tagged as such. # Each extraction is called phrase#, where phrase T. Next, each phrase is paired with each positive-adjectivei Vocabpositive, and each negative-adjectivej Vocabnegative and then we search each of these pairs using each engine in the Search API, which returns the amount of hits for each pair. The figure below shows the final implementation of the SA.

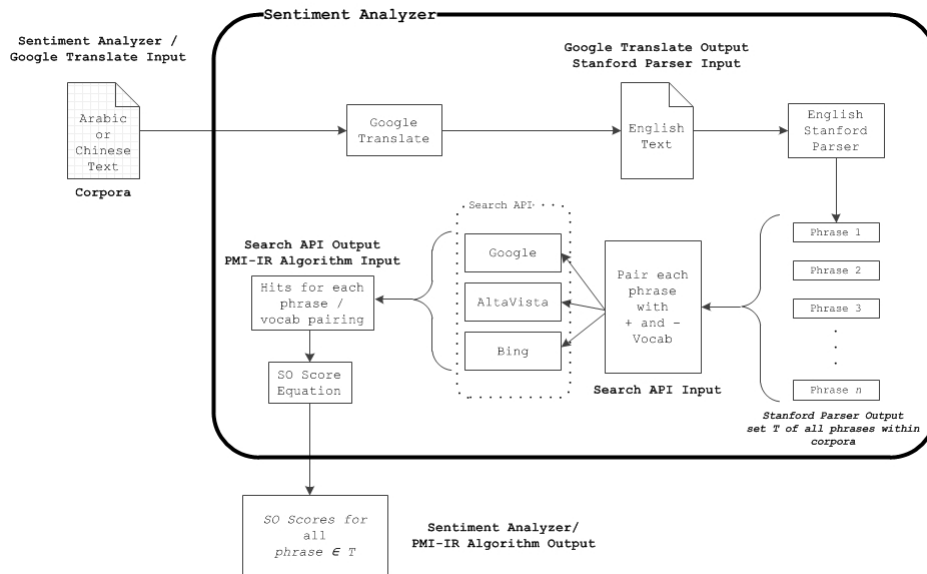


Figure 8: Final implementation of the SA

3.4 RESULTS

After the SO measure for each phrase was calculated per search engine, we classified the phrases individually as positive, neutral or negative to see how well the solution performed. This was done using a five judge panel. We then calculated the manual classification by doing a majority vote on the individual ones.

Turney [14] classified all data with an SO score > 0 as “Thumbs Up” (positive) or “Thumbs Down” (negative) < 0 . In our work, we also decided to classify neutral phrases, which means that we needed two thresholds: one upper and one lower.

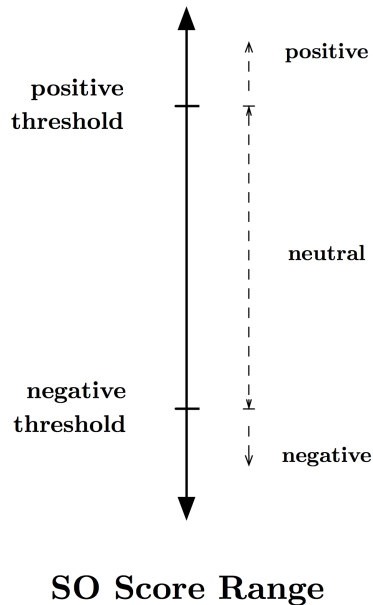


Figure 9: SO score range

This is not as trivial as putting the thresholds for positive phrases above an SO score of 3.2, and negative phrases below 2.56. We decided to calculate the thresholds by comparing the SO score of each phrase to the manual classification, and then loop through all possible combinations of thresholds to find the optimal thresholds, based on accuracy, for assigning negative and positive phrases. Phrases with an SO score between the negative and positive thresholds are classified as neutral.

As mentioned previously, Turney analyzed the sentiment of online reviews where the author indicated their sentiment using a 5 star rating scale, and he was able to verify the PMI-IR algorithm’s accuracy using those ratings. Since

our corpora consisted mostly of fictional literature and news articles, we needed a different means of calculating the SA's accuracy. Five judges read the 1000+ phrases extracted from the corpora and assigned a value of 1 for a positive phrase, -1 for negative, and 0 for neutral. These values were simple indicators of polarity and held on actual weight (i.e they could have been 1,2,3). We then took the majority rule of the five judges to determine the actual classification of a phrase. The judges scored an example phrase from the corpora, "old clock", with $\{-1, -1, 1, 1, -1\}$. Since 3 negatives $>$ 2 positives, the actual classification of the phrase was said to be negative.

The accuracies we found are presented in . Although the accuracies for both languages do not seem to be very high, it is important to note that our implementation not only attempts to classify positive and negative phrases (as Turney did), but neutral as well. Thus, creating a greater margin of error.

Contrary to findings by Turney [14], the NEAR operator was not much more accurate than Google and AV's (implicit AND) operator.

The confusion matrix below shows how accurate our SA was at classifying phrases within Chinese texts using Google as search engine. We see that 120 actual positive phrases (i.e. classified manually as positive by us) were classified as positive by our solution. We also see that 60 positive phrases were classified as negative etc.

The recall value of 63% for actual positive phrases shows us that there were 120 out of 191 were predicted as positive. The precision value of positive phrases shows us how many of these predictions were correct.

The confusion matrices show us the percentage of positive, negative, and neutral we received from each of the search engine in our Search API. Appendix 7 contains the full results of our implementation.

4 DISCUSSION

To analyze a writer’s sentiment, the best-case scenario would be directly asking the author, “Were you writing in a positive or negative tone?”. The worst-case scenario: the inability to read the author’s text or having absolutely no idea of the author’s tone. The middle-ground: analyzing the text either manually or by machine. Manual sentiment analysis done by humans tends to be more accurate than sentiment analysis done by SAs because humans are, as French and Labiouse [3] point out, better at answering subcognitive (cultural and perceptual) questions; however, SAs are highly efficient and can process large sets of text.

Our solution wraps modern elements in the field of sentiment analysis into a simple package. A model like the one presented is advantageous because of its scalability and flexibility. While the final model of our SA interfaces with Chinese and Arabic, it could also interface with 49 other languages. And as GT expands, so will our SA.

On the other hand, our IR process (performed by the Search API in our model) is particularly inefficient. Retrieving hits for one phrase in the corpora took between 1.5-7 minutes# and the corpora contained over 1000 phrases, so that’s a total of roughly 25-115 hours spent simply retrieving hits. One possible way to speed up this process would be to distribute the searches on multiple computers or use a local database.

4.1 POSSIBLE CAUSE FOR LOW ACCURACY

Although the SA attempted to find not only positive and negative, but neutral phrases as well, we hypothesized and expected greater accuracy than we achieved.

Pang and Lee [8] point out that one of the largest applications of sentiment analysis is processing review-related websites. Turney’s [14] corpora came from <http://epinions.com>. Siquiera and Barros [9] modified the feature extraction part of Turney’s algorithm and achieved slightly better results than Turney. The corpora they used came from <http://e-bit.com.br>, a site dedicated to collecting reviews of online stores. Again, a corpora comprised of reviews.

The premise behind a review is this: “Did you or didn’t you like product o service?” People who have a “so-so”, or neutral, view of a product, aren’t typically the same people writing the reviews. Reviews are generally for those who really love or really disliked a product or service. Because of this, reviews tend to be more emotional than fictional literature or news, which our corpora is made of. For instance, consider the review pulled from <http://epinions.com>, which contains phrases such as “bad writing” and “this garbage”.

The more emotional a phrase, the farther its SO score would be from the neutral threshold, and the more likely it would be manually classified as either very positive or negative. Simply put, neutral phrases are both difficult for an SA as well as a human to classify, and more often negatively affect an SAs accuracy.

The sentiment of the review (seen in the phrases “this garbage”, “bad writing”) closely matches the rating (1 star). It would be interesting to see how accurate the SA would be at analyzing Chinese and Arabic reviews.

Another possible reason for the low accuracy might have to do with the content of some of the texts, especially the ones written in Arabic. A majority of the texts came from an online Arabic news source, and some of the words extracted were “Arab”, “Israel”, “Palestine”, etc. Because the PMI-IR algorithm uses IR from the web, if a word has a “bad name” on the web, that means it will typically co-occur on web-pages with other negative words, and thus, have a low SO score. Tweetfeel.com, a website that claims “real-time Twitter search with feelings using insanely complex sentiment analysis” rated each of the three aforementioned words as 75-85% negative. The five judge panel that manually rated the phrases typically gave phrases with words like these neutral scores. Our SA judged both of the phrases “Arab organization” and “Israeli writer” as negative.

French and Labiouse [3] used the PMI-IR algorithm in an attempt to classify the word “lawyers”. In essence, they asked the PMI-IR algorithm to classify lawyers as either computers, cats, telephones, slimeballs, bastards, kangaroos, or robins and what they found was astonishing. Turney’s algorithm judged lawyers to be least like slimeballs and bastards and most like computers, cats, and telephones- a judgement that differs from the general public. This shows that machine learning algorithms such as PMI-IR have little sense of today’s cultural and political climate, more specifically, have little idea that there is a conflict between Israel and Palestine and Arabs and Jews that is widely spoken about negatively on the web.

Lastly, there’s another issue. In translating the corpora from its natural language to English, we feel as if some informal words may have been mistranslated, and thus, lost their meaning altogether. An example extracted translated phrase from one of the Chinese texts, “fished apes”, makes little sense. Perhaps the meaning in Chinese is comprehensible, but not in English. Earlier we presented the question: Is enough of the author’s sentiment captured during translation? It seems as if the answer to this question is “No”.

4.2 APPLICATIONS

There are a variety of possible applications for sentiment analyzers that process multiple languages like the one presented in this report. International enterprises often ask their employees to write reviews about the company (workplace conditions, etc.). If this enterprise employed thousands of people who spoke many different languages, it would have to process thousands of reviews in many languages. One possible application would be for the enterprise to run all of these reviews through an SA such as the one presented in this report. The enterprise could quickly and nearly instantly receive a score of how their employees rated the company. Our implementation can analyze any language GT offers (currently 51 different languages) and has the ability to detect the source language.

4.3 POSSIBLE IMPROVEMENTS

Though research has shown [14] adjective paired with nouns to be efficient in capturing sentiment, some have used parts of speech such as verbs and adverbs like “softly” or “horribly”. Our model could be expanded to include these tags to increase accuracy.

5 CONCLUSION

As explained throughout this report, a sentiment analyzer accepts a text as its input and returns a sentiment score. This sentiment score, in theory, is the key that unlocks the door to the author's tone during authorship of the text inputted to the sentiment analyzer. During this project, we set out to build a sentiment analyzer that was: (1) able to do sentiment analysis of Chinese and Arabic texts, (2) was built upon state-of-the-art methods and technologies, and was (3) both efficient and accurate.

Our implementation worked as follows: (1) the Chinese and Arabic texts were translated to English using Google Translate, then (2) the phrases (adjective and noun) were extracted from the translated text using the Stanford Parser, and (3) the sentiment analyzer combined the phrases from the translated text and each word in the set of polar (strongly negative or positive) vocabulary words, and then (4) used the search engines: Google, AltaVista and Bing and retrieved the information needed to calculate the semantic orientation score for each phrase within the texts.

Although the SA attempted to find not only positive and negative, but neutral phrases as well, we hypothesized and expected greater accuracy than we achieved. Our proposed solution has elements that differ from existing solutions. Turney [14] performed sentiment analysis on English text only, we have carried out semantic orientation analysis on Arabic and Chinese text. We also introduced classifying phrases as neutral simply because some phrases did not have a negative or positive tone.

A major plus of this application is its usefulness and it can be extended and improved. We concentrated our work on the two languages Arabic and Chinese, but the sentiment analyzer that we have implemented can be used for any language that Google Translate offers currently. In future work, we aim to: limit the Information Retrieval process (i.e. searching for phrases on Google, AltaVista and Bing) to a specific time interval, for example only gather information that is a month or a year old, enabling us to obtain another dimension of information, namely current semantic orientation trends.

6 REFERENCES

References

- [1] Das, S., & Chen, M. "Yahoo ! for Amazon : Extracting Market Sentiment from Stock Message Boards". *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*. 35, 43, 2001.
- [2] Firth, J.R. *A Synopsis of Linguistic Theory 1930-1955 in Studies in Linguistic Analysis*. Location: Oxford: Philological Society, pp. 1-32. . 1957.
- [3] French, R. M. and Labiouse, C. "Why co-occurrence information alone is not sufficient to answer subcognitive questions". *Journal of Theoretical and Experimental Artificial Intelligence*, 13(4), 419-42. 2001.
- [4] Khoja, S. "APT: Arabic Part-of-speech Tagger". *Proceedings of the Student Workshop at the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL2001)*. 2001.
- [5] Klein, D., and Manning, C. D. "Fast exact inference with a factored model for natural language parsing". *Advances in Neural Information Processing Systems* 15. 2003.
- [6] LI Juan, ZHANG Quan, JIA Ning, Semantic orientation identification for Chinese opinion terms, in *Computer Engineering and Applications*, 45(2), pp. 131-133. 2009.
- [7] Ma X, Zhang X, Zhao H, Lu B-liang. "Dependency Parser for Chinese Constituent Parsing". *CIPS-SIGHAN Joint Conference on Chinese Language Processing* 2010.
- [8] Pang B, Lee L. "Opinion Mining and Sentiment Analysis". *Foundations and Trends in Information Retrieval* 2(1-2), pp. 1????? ERROR HERE MUST FIX !!!! 135, 2008.
- [9] Siqueira H, Barros F. A "Feature Extraction Process for Sentiment Analysis of Opinions on Services". *III International Workshop on Web and Text Intelligence (WTI - 2010)*.
- [10] Tong, R. M. "An Operational System for Detecting and Tracking Opinions in Online Discussion". *Proceedings of the Workshop on Operational Text Classification (OTC)*. 35. 2001.
- [11] Tsytsarau, M, Palpanas, T. "Mining subjective data on the web". *Data Mining and Knowledge Discovery*. 2010.
- [12] Turney, P. D, Littman, M. L. "Measuring praise and criticism: Inference of semantic orientation from association". *ACM Transactions on Information Systems (TOIS) Volume 21 Issue 4, October 2003*.

-
- [13] Turney, P. D. "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL". *Lecture Notes in Computer Science. Pages 2167: 491-502, 2001.*
- [14] Turney, P. D. "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews", *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, July 07-12, 2002.*
- [15] Weber, G. "The world's ten most influential languages". Internet: <http://www.andaman.org/BOOK/reprints/weber/rep-weber.htm>. May 6, 2008. [Nov. 17, 2010].
- [16] Zhendong DONG, Qiang DONG. "HowNet". Internet: http://www.keenage.com/zhiwang/e_zhiwang.html [Nov. 15, 2010].
- [17] ZHU Yan-lan, MIN Jin, ZHOU Ya-qian, HUANG Xuan-jing, WU Li-de, "Semantic Orientation Computing Based on HowNet". *Journal of Chinese Information Processing, Vol. 20,.* 2006.

7 APPENDICES

Glossary & Abbreviations

Table of Abbreviations

Abbreviation	Meaning
POS	part-of-speech
NN	Noun POS tag
JJ	Adjective POS tag
NLP	Natural Language Processing
PMI-IR	Pointwise Mutual Information Information Retrieval
SA	Sentiment analyzer
SO	SO Semantic orientation
API	Application programming interface
GT	Google Translate
AV	AV AltaVista

Table 2: Table of Abbreviations

Glossary

Sentiment analysis The process of analyzing a text or document to determine the overall tone or feeling.

Semantic orientation The polarity of a word or phrase (positively or negatively oriented).

Machine translation A translation of a text or document done by software from one language to another.

Natural Language Processing (NLP) Making use of computers to interpret and manipulate words as part of a language

Part-of-Speech Tagger A part-of-speech tagger is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc.

Noun A part of speech, or word, that refers to a person, place, thing, event, idea, substance, etc.

Adjective A part of speech, or word, that describes or qualifies a noun.

Adverb A part of speech, or word, that describes a verb or an adjective.

Verb A part of speech, or word, that implies an action, e.g. “play”, “are”.

TOOLS

Stanford Parser <http://nlp.stanford.edu/software/lex-parser.shtml>

Google Translate <http://translate.google.com/>

Google <http://www.google.com>

Twitter <http://twitter.com/>

Bing <http://www.bing.com>

AltaVista <http://www.av.com>

RESULTS

Search engine	Accuracy Chinese	Accuracy Arabic
Google	44%	45%
AltaVista	46%	45%
Bing	46%	49%

Table 3: Accuracy per search engine per language

		PREDICTED				
		<i>Positive</i>	<i>Negative</i>	<i>Neutral</i>	Recall	Precision
ACTUAL	<i>Positive</i>	37	35	108	21%	43%
	<i>Negative</i>	16	53	69	38%	38%
	<i>Neutral</i>	33	51	163	66%	48%

Table 4: Confusion matrix and recall & precision for Arabic using Google

		PREDICTED				
		<i>Positive</i>	<i>Negative</i>	<i>Neutral</i>	Recall	Precision
ACTUAL	<i>Positive</i>	55	30	86	31%	49%
	<i>Negative</i>	7	45	21	33%	39%
	<i>Neutral</i>	51	41	155	63%	46%

Table 5: Confusion matrix and recall & precision for Arabic using AltaVista

		PREDICTED				
		<i>Positive</i>	<i>Negative</i>	<i>Neutral</i>	Recall	Precision
ACTUAL	<i>Positive</i>	68	38	74	38%	54%
	<i>Negative</i>	13	64	61	46%	41%
	<i>Neutral</i>	46	54	147	60%	52%

Table 6: Confusion matrix and recall & precision for Arabic using Bing

		PREDICTED				
		<i>Positive</i>	<i>Negative</i>	<i>Neutral</i>	Recall	Precision
ACTUAL	<i>Positive</i>	120	60	11	63%	45%
	<i>Negative</i>	59	84	10	55%	43%
	<i>Neutral</i>	46	53	17	11%	45%

Table 7: Confusion matrix and recall & precision for Chinese using Google

		PREDICTED			Recall	Precision
		<i>Positive</i>	<i>Negative</i>	<i>Neutral</i>		
ACTUAL	<i>Positive</i>	138	46	7	72%	47%
	<i>Negative</i>	65	80	8	52%	45%
	<i>Neutral</i>	92	51	14	9%	48%

Table 8: Confusion matrix and recall & precision for Chinese using AltaVista

		PREDICTED			Recall	Precision
		<i>Positive</i>	<i>Negative</i>	<i>Neutral</i>		
ACTUAL	<i>Positive</i>	98	82	11	51%	57%
	<i>Negative</i>	25	115	13	75%	41%
	<i>Neutral</i>	52	86	19	12%	44%

Table 9: Confusion matrix and recall & precision for Chinese using Bing