Identification of Certain Emotions in Text

Pranjal Singh
Divyanshu Bhartiya

Advisor: Prof. Amitabha Mukerjee
Deptt. of Computer Science & Engg.
IIT Kanpur, India
{spranjal,divbhar,amit}@iitk.ac.in

April 12, 2012

Abstract

Emotions have been widely studied in psychology and behavior sciences, as they are an important element of human nature. In this project we aim to identify certain basic emotions present in a text. We have used ISEAR corpus which contains sentences related to certain emotions, a PCFG parser and determind the weight of a sentiment and then tried to give weight to context with the help of PLSA algorithm which is basically a probabilistic method to determine the meaning behind the words. The results of the project are quite promising.

1 Introduction

Emotions play an important role in domain of human intelligence, rational decision making, communication, perception and much more. In the past 10 years, this topic has hit almost every researcher related to Artificial Intelligence and a lot of researches have been done on emotion recognition, whether it is speech-emotion recognition or facial-emotion recognition. At the same time, textual emotion recognition is increasingly attracting attention.

The emotion analysis on sentence level may also be important for more detailed emotion analysis systems. Previos works have focussed mainly on areas such as blogs, news, text messages,etc. which has benefitted the areas such as Computational Linguistics, Online Counselling, Social-Networking sites, Markets, etc.

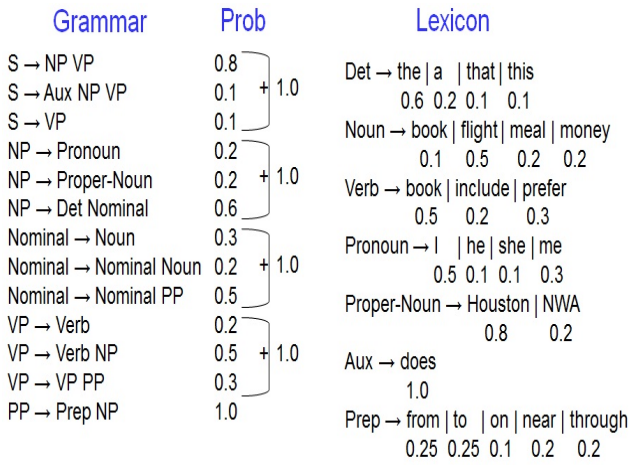
This report describes experiments concerned with the emotion analysis of self given sentences which are analyzed by a supervised learning approach using tagged dictionary and unsupervised learning approach using ISEAR[1] database. In this work, some of the hardest problems involve acquiring large corpora tagged with detail linguistic expressions that indicate emotion and training time of the data-set.
The remainder of this paper is organized as follows. Section 2 describes the basics of parsing using PCFG parser,an important part in training machine. Section 3 presents the concepts of PLSA method. Section 4 presents the Implementation Details followed by the Results in Section 5.

2 Parsing: PCFG(Probabilistic Context Free Grammar)

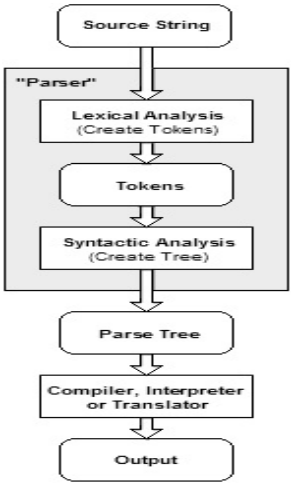
In computer science and linguistics, parsing, or, more formally, syntactic analysis, is the process of analyzing a text, made of a sequence of tokens (for example, words), to determine its grammatical structure with respect to a given (more or less) formal grammar. Parsing can also be used as a linguistic term, for instance when discussing how phrases are divided up in garden path sentences.

In computing, a parser is one of the components in an interpreter or compiler that checks for correct syntax and builds a data structure (often some kind of parse tree, abstract syntax tree or other hierarchical structure) implicit in the input tokens. The parser often uses a separate lexical analyser to create tokens from the sequence of input characters. Parsers may be programmed by hand or may be (semi-)automatically generated (in some programming languages) by a tool.

A PCFG[2] is a probabilistic version of a CFG where each production has a probability.
Probabilities of all productions rewriting a given non-terminal must add to 1, defining a distribution for each non-terminal. String generation is now probabilistic where production probabilities are used to non-deterministically select a production for rewriting a given non-terminal.



Source:Google Images



Source:Wikipedia

He might have lung cancer. It s just a rumor ... but it makes sense. He is very depressed and that s just the beginning of things

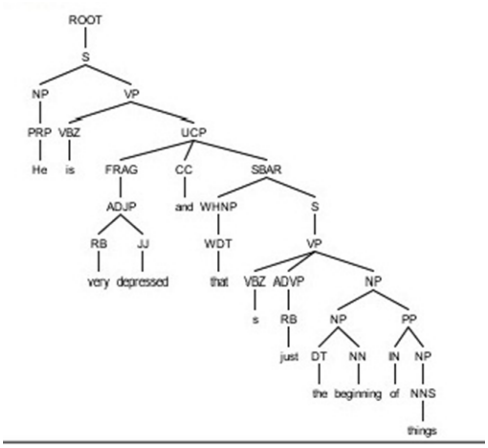


Fig. PCFG Parser Result

3 PLSA:Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis(pLSA)[4] is a technique from the category of topic models. Its main goal is to model cooccurrence information under a probabilistic framework in order to discover the underlying semantic structure of the data.
For this particular application, our training data is a corpus—a large set of documents—that is usually represented in the form of a document-term matrix (this indicates the number of times each word appears in each document). The goal of pLSA is to use this co-occurrence matrix to extract the so-called "topics" and explain the documents as a mixture of them. PLSA considers that our data can be expressed in terms of 3 sets of variables:

- Documents: d $\varepsilon D = \{d_1, ..., d_N\}$ observed variables. Let N be their number, defined by the size of our given corpus.
- Words: w $\varepsilon W = \{w_1, ..., w_M\}$ observed variables. Let M be the number of distinct words from the corpus.
- Topics: z $\varepsilon Z = \{z_1, ..., z_K\}$ latent(or hidden) variables. Their number, K, has to be specified a priori.

A generative process for the documents is:

- First we select a document d_n with probability $P(d)$.
- For each word $w_i, i\varepsilon\{1, ..., N_w\}$ in the document d_n:
  - Select a topic z_i from a multinomial conditioned on the given document with probability $P(z|d_n)$.
  - Select a word w_i from a multinomial conditioned on the previous chosen topic with probability $P(w|z_i)$.

There are some important assumptions made by the presented model: A generative process for the documents is:

- Bag-of-words. Intuitively, each documentis regarded as an unordered collection of words. More precisely, this means that the joint variable (d,w) is independently sampled and, consequently, the joint distribution of the observed data will factorize as a product:
$$\text{P(D,W)} = \prod_{(d,w)} P(d, w)$$
- Conditional independence. This means that words and documents are conditionally independent given the topic: $P(w|d, z) = P(w|z)P(d|z)$ or $P(w|d, z) = P(w|z)$. (This can be easily proved by using d-separation into our graphical model: the path from d to w is blocked by z.)

The predictive probability of pLSA mixture model is denoted by $P(w|d)$, so the objective function is given by the following expression:
$\text{L} = \prod_{(d,w)} P(w|d) = \prod_{(d\varepsilon D)} \prod_{(w\varepsilon W)} P(w|d)^{n(d,w)}$

where $n(d|w)$ represents the observed frequencies, the number of times word w appears in document d. This is a non-convex optimization problem and it can be solved by using Expectation-Maximization (EM) algorithm for the log-likelihood:

$$\mathbf{M} = \log L = \prod_{(d\varepsilon D)} \prod_{(w\varepsilon W)} n(d, w). \log \sum_{z\varepsilon Z} P(w|z)P(z|d)$$

4 Implementation Details

Implementation was carried out in two stages: Supervised learning and Unsupervised Learning. Supervised learning[3] involves a corpus with words being annotated with an emotion tag along with an itensity. This phase of implementation involves first parsing the sentences into tokens , as the words are stored in corpus arranged by their part of speech tag. The tag includes the emotion category and intensity. The tokenized sentence is then filtered of the identifiers,determiners and words commonly called the "stop words". The resulting set of words is then searched for in the corpus, to get the necessary sentiment(positive or negative or neutral) with a score. This implementation does not consider semantics and ignores the context.

Unsupervised Learning[4] tries to include the semantics of the sentence but without any prior knowledge(category). This approach is carried by first training the machine. We used **Probabilistic Latent Semantic Analysis** to achieve unsupervised learning. We have used ISEAR database to train the machine, created a term by document matrix out of the 7500 sentences after removing the stop words. The PLSA analyses the data and assigns probabilities(scores) to the documents and words by the specific categories. It can then return the probability of the document to be in the seven categories it defined. The PLSA returns a cluster of words for each category, which are not emotion specific. Hence we can only make a rough estimate of the emotion by the words. Using a large corpora can solve this problem.

The supervised learning is grammar specific,fast but it returns only the sentiment, not the emotion as each sentiment is subdivided into many subcategories. Hence rough classification is not possible. On the other hand, PLSA takes into account context, looks at semantics but training requires a large corpus.

5 Results

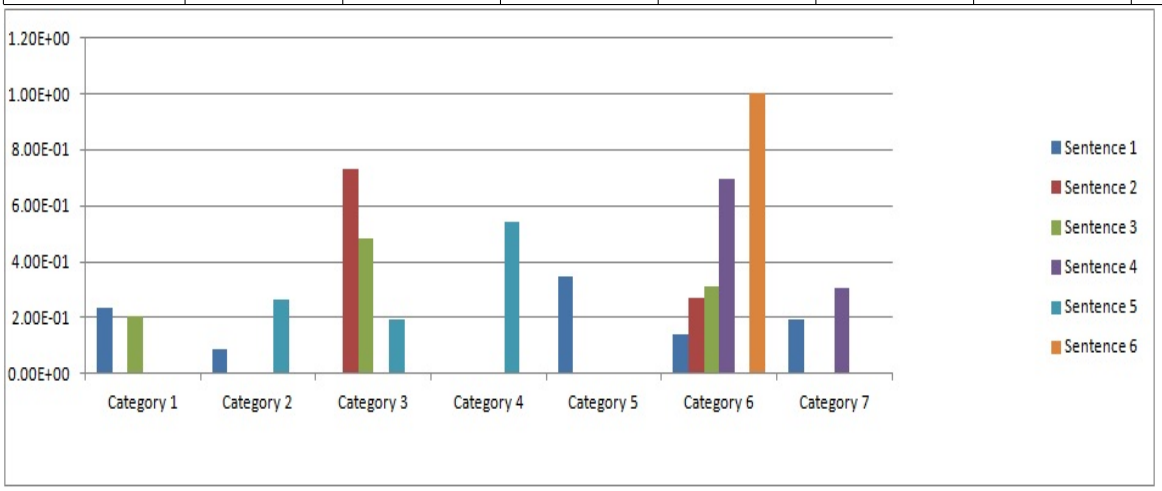
As mentioned in Section 4, here are few results on some sentences mentioned in Carlos Strapparava[3] paper. Following are the results of Supervised Learning:

SENTENCES	TYPE	SCORE
I am so angry. She cannot get work off for his his show on 30th, and were stuck in traffic for almost 3 hours today,preventing us from seeing them,bastards.	Negative	-0.429
It is time to snap out of this. It is time to pull things together. This is ridiculous. I am going nowhere. I am doing nothing.	Negative	-0.336
He might have lung cancer. It is just a rumour but it makes sense. He is very depressed and that is just the beginning of things.	Negatice	-0.357
This week has been the best week I have had since I cannot remember when. I have been so hyper all week, it has been awesome.	Positive	0.0306
Oh and a girl from my old best school got run over and died the other day which is horrible, especially as it was a very small village school so everybody knew her.	Negative	-0.069
French men shake your hand as they say good morning to you. This is a little shocking to us fragile Americans,who are used to waving to each other in greeting.	Negative	-0.058

The above results present quite an accuracy to the sentiments. The score can't be an good way to judge things because it depends on the listener's perspective too, although the score is a good measure.

The unsupervised learning gave us 7 categories on the corpus we trained it on. The corpus caontained sentences on anger, disgust, fear, joy, guilt, sadness,and shame. The PLSA when trained for 7 categories returned a cluster of words with maximum probability for each category. These are some of the top probable words for each category.
Category 1 :'work', 'family', 'sad'
Category 2 :'exam', 'failed', 'university'
Category 3 :'close', 'died', 'person'
Category 4 :'people', 'met', 'relationship'
Category 5 :'asked', 'party', 'give'
Category 6 :'angry', 'good', 'guilty',
Category 7 :'disgusted', 'afraid', 'fear'

Sen/Categ.	Categ. 1	Categ. 2	Categ. 3	Categ. 4	Categ. 5	Categ. 6	Categ. 7
Sentence 1	2.36E-001	8.86E-002	3.51E-081	1.76E-248	3.45E-001	1.38E-001	1.91E-001
Sentence 2	1.41E-169	2.54E-140	7.32E-001	4.07E-189	4.13E-223	2.68E-001	2.12E-251
Sentence 3	2.05E-001	1.41E-164	4.83E-001	6.62E-252	2.19E-151	3.11E-001	0.00E+000
Sentence 4	1.87E-115	0.00E+000	0.00E+000	0.00E+000	9.29E-104	6.97E-001	3.02E-001
Sentence 5	9.92E-313	2.65E-001	1.92E-001	5.41E-001	5.11E-074	1.47E-219	3.14E-142
Sentence 6	0.00E+000	0.00E+000	3.25E-268	0.00E+000	0.00E+000	1.00E+000	2.56E-142



Though it is difficult to guess the emotion from the cluster obtained , yet it can be helpful since the words depict fairly a sense.

6 Further Work

Classification of emotions seems more difficult than sentiment classification. Emotional words and phrases play important role for emotion recognition, but more linguistic expressions such as negative words, conjunctions, punctuations should be considered for more accurate recognition. The amount of time that the code requires can be reduced. To improve the quality of results, the training should be done with a large corpus as clearly visible in Unsupervised Learning that the cluster of words can be made more accurate by using a large corpus. Also a mixed approach of supervised and unsupervised learning can be brought to use.

7 Acknowledgement

We thank Prof. Amitabha Mukerjee for his valuable support throughout the project, guiding us from time to time and looking into the project when it was needed. We have used open source available "Alchemy API", PLSA code from "Mathieu's blog" and textmining modules in python, and ISEAR database. We acknowledge you for you support.

References

[1] Kim, Sunghwan Mac and Valitutti, Alessandro and Calvo, Rafael A.:Evaluation of unsupervised emotion models to textual affect recognition, Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, CAAGET '10 California

[2] Xu Zhe and Anthony, Boucouvalas: Text-to-Emotion Engine for Real Time Internet Communication, International Symposium on CSNDSP 2002

[3] Carlos Strapparava and Rada Mihalcea: Learning to Identify Emotions in Text SAC '08 Proceedings of the 2008 ACM symposium on Applied computing ACM New York, NY, USA, 2008

[4] Changqin Quan and Fuji Ren: An Exploration of Features for Recognizing Word Emotion COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics, 2010