The Unified Sentiment Lexicon using GPUs

Liliana Ibeth Barbosa-Santillán ibarbosa@cucea.udg.mx University of Guadalajara, México
Inmaculada Alvarez de Mon y Rego inmaculada.alvarezdemon@upm.es Technical University of Madrid, Spain

Introduction

This approach aims at aligning, unifying and expanding the set of sentiment lexicons which are available on the web in order to increase their robustness of coverage. A sentiment lexicon is a critical and essential resource for tagging subjective corpora on the web or elsewhere. In many situations, the multilingual property of the sentiment lexicon is important because the writer is using two languages alternately in the same text, message or post.

Our USL approach computes the unified strength of polarity of each lexical entry based on the Pearson correlation coefficient which measures how correlated lexical entries are with a value between 1 and -1, where 1 indicates that the lexical entries are perfectly correlated, 0 indicates no correlation, and -1 means they are perfectly inversely correlated and the UnifiedMetrics procedure for CPU and GPU, respectively.

The Main USL approach

```
(1) procedure UnifiedSentimentLexicon(seeds)
(2)   SentimentLexicons ← FocusCrawlerEngine(seeds);
(3)   Clusters ← SelectorLanguages(SentimentLexicons);
(4)   for i ← 1, NumberofClusters do
(5)     for j ← 1, NumberofSentimentLexicons do
(6)       for k ← 1, NumberofLexicalEntries do
(7)         if MetricSearcher(LexicalEntry(k)) = 1 then
(8)           MetricsLexicon (j) (k) ← LexicalEntry(k);
(9)         else
(10)          NoMetricsLexicon(j)(k) ← LexicalEntry(k);
(11)        end if
(12)        if MetricTransformer(NoMetricsLexicon(j)(k))≥ 0 then
(13)          MetricsLexicon(j)(k) ← NoMetricsLexicon(j)(k);
(14)        end if
(15)      end for
(16)    end for
(17)  end for
(18)  if SentimentLexiconIntersection(Clusters(MetricsLexicon)) = 1 then
(19)    IntersectionLexicalEntries(j)(k) ← Clusters(MetricsLexicon);
(20)  else
(21)    NoIntersectionLexicalEntries(j)(k) ← Clusters(MetricsLexicon);
(22)  end if
(23)  NoIntersectionLexicalEntries(j)(k) ← LexicalEntriesSubstracter
(24)  (SentimentLexicon, IntersectionLexicalEntries);
(25)  n ← LexicalEntriesDivisor(IntersectionLexicalEntries, NumberofCoresGPU);
(26)  for i ← 1, NumberofClusters do
(27)    for j ← 1, n do
(28)      r ← UnifiedMetrics_CPU(Cluster(i)(SentimentLexicons));
(29)      USL(i) ← UnifiedMetrics_GPU(n, r, Cluster(i)(IntersectionLexicalEntries));
(30)      UnifiedSentimentLexicon ← UnionSentimentLexiconEngine(j);
(31)    end for
(32)  end for
(33)  OntoLexicon ← Lexicon2OntologyConverter(UnifiedSentimentLexicon);
(34) end procedure
```

Acknowledgments

The authors are grateful to CONACYT for funding this research project and NVIDIA for donation cards. This works was published in the Journal "Mathematical Problems in Engineering".

Model

Our approach calculates the Pearson correlation score between each sentiment lexicon and the USL by obtaining as many constants as there are sentiment lexicons in the cluster. For example, if the cluster belongs to the English language, then there are four constants that fall into each sentiment lexicon, as shown in the Pearson correlation set *PearsonCorrelation = p1,p2,p3,p4*. This calculation is performed only once and executed by the CPU.

Since the number of lexical entries is high, the computation of the USL score should be divided into several coprocessors (cores) in order to accelerate the process. In fact, each coprocessors of the GPU has as an input: (a) the strength of polarity of n lexical entries and (b) the vector with Pearson values. Each coprocessor computes the strength of polarity of every lexical entry until there are no lexical entries left. The score for each lexical entry is multiplied by the Pearson correlation between all the sentiment lexicons. Consider

$$\alpha_i = \mathbf{p}_1 * \mathbf{v}_1 \quad (1)$$
$$\beta_i = \mathbf{p}_2 * \mathbf{w}_i$$
$$\gamma_i = \mathbf{p}_3 * \mathbf{y}_i$$
$$\delta_i = \mathbf{p}_4 * \mathbf{z}_i$$

In addition, USL performs a total of subjectivity sums. Consider

$$\varepsilon_i = \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i + \boldsymbol{\gamma}_i + \boldsymbol{\delta}_i \quad (2)$$

The USL score is normalized by dividing the total number of subjectivity for each lexical entry by the Pearson correlation sum of the lexical entries that were assessed $\zeta = p_1 + p_2 + p_3 + p_4$, as follows:

$$USL_1 = \frac{\varepsilon_1}{\zeta_1} \quad (3)$$

The GPU results are the lexical entries combined with the USL score (these are input by the CPU). Their main function is to join all the partial results in the USL.

Finally, the CPU transforms the USL into an ontology called OntoLexicon in OWL language. The pseudocode of the main USL approach is shown in the main USL approach.

Rate of Lexical entries total by sentiment lexicon

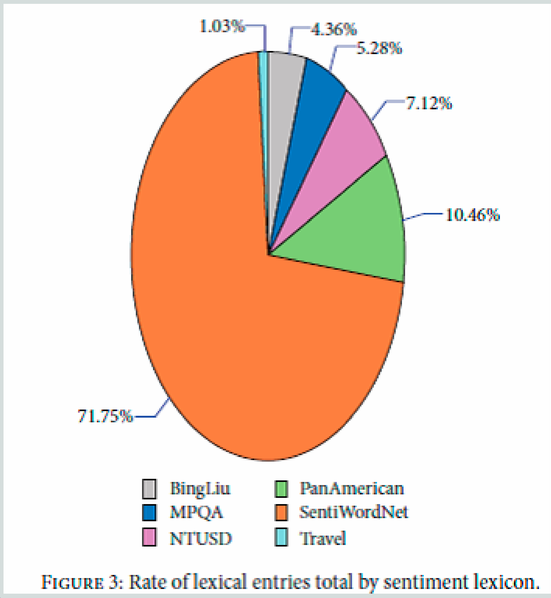


FIGURE 3: Rate of lexical entries total by sentiment lexicon.

Research Questions

Q1. Is it possible to unify the sentiment lexicons available on the web and align and expand them automatically?

Q2. Is it possible to transform a Unified Sentiment Lexicon into a generative lexicon based on a core ontology?

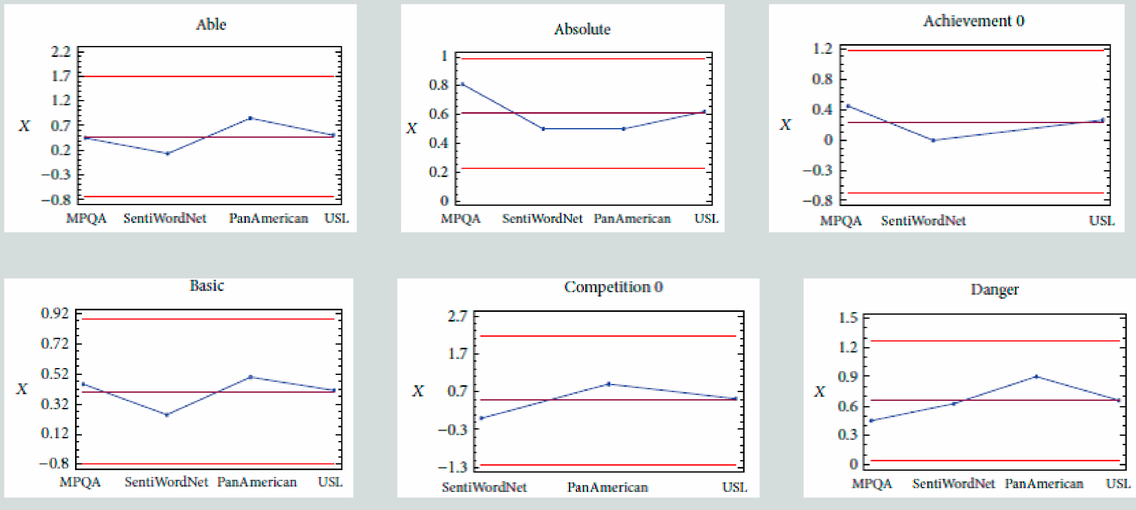
References

[1] A. Trujillo. Automating the lexicon: Research and practice in a multilingual enviroment. In *Natural Languages Engineering, vol. 2, no. 3, pp. 277-285, 1996.*

[2] E. Lindholm, J. Nickolls, S. Oberman and J. Montrym. NVIDIA Tesla: a unified graphics and computing architecture. In *IEEE Micro, vol. 28, no. 2, pp. 39-55, 2008.*

Results

For the first cluster - English - a subset of lexical entries is shown in the next figures.



The results are quite satisfactory although some minor problems have been detected. These problems are mainly due to the existence of expressions that can have both a positive and a negative value, and only one of the values is signalled. In the subset analysed, for instance, that is the case of the word basic, which can sometimes have a negative value when it is used to refer to the attributes or properties of an object or to the quality or level as in "the hotel room was too basic". Another problem is the influence of the results of considering all the lexicons for the final result. That is what happens in the case of the word achievement.