iJournals: International Journal of Software & Hardware Research in Engineering
ISSN:2347-4890, ijournals.in/ijshre
Volume 13 Issue 1, January 2025

An Alternate Formulation of The Mutual Information Statistic Which Yields a More Realistic Measure, Leading to a More Precise and Dependable Model of Natural Language Translation Probabilities Among Parallel Corpora

Author: Parthasarathy Srinivasan

Beehive Software Solutions

ospark2477@gmail.com

DOI: 10.26821/IJSHRE.13.01.2025.130104

ABSTRACT

The Mutual Information is an important statistic which quantifies the relation between two probability distributions in terms of the mutual information content across the two (parallel) linguistic corpuses under consideration. This is a significant conceptual model which forms the basis of techniques which explore the correspondence of the parallel corpuses for similarity and translation. The traditional formulation of Mutual Information is useful in this regard to model parallel corpus instances and obtain inferences useful for corresponding translations. However, the inherent lacuna in this scheme is the loss of generality due to reliance on specific corpus instances and thereby failing/underperforming with respect to realistic precision and reliability of the inferred model parameters for general use. This work is an attempt to address this lacuna by introducing a novel formulation for (True) Mutual Information which is robust from being affected by the loss of generality (condition) mentioned above. The novel formulation recognizes the fact that the (True) MI statistic can be obtained as measure of deviation of the individual probabilities of the distribution from critical values (as will be elaborated) , which are obtained by computing the Takens embedding of the distribution. The resultant measure is robust in generality as the Takens embedding captures how the distribution tends to move (in time), thus providing a Truer picture which is independent of the specifics of the choice of particular corpus instances..

Keywords: Mutual Information, Machine Translation, Parallel Corpus, Natural Language, Probabilities, Model Parameters.

1. INTRODUCTION

Automatic (machine based) translation of documents from one language to another has been a subject of active study since the 1990's. One particularly insightful approach is the modelling of the translation process as a (translation) probability distribution obtained by aligning the words across two parallel corpuses. In this approach , as detailed in [1] , the authors employ a technique known as the EM algorithm which essentially models the translation probabilities as counts of appearances of linguistically aligned words in instances of chosen parallel corpora.

The algorithm then iterates to update the translation probabilities in steps by using a summation of the counts (of each of the corpora pairs) obtained as

above as the new value of the translation probability. The details of the algorithm and its theoretical basis are described in Section 4.1 of [1]. While this algorithm obtains effective values of translation probabilities, it can be easily visualized that the algorithm is heavily dependent on the choice of the corpora chosen , in obtaining the translation probabilities of the constituent words of the corpora. The convergence of the algorithm is also subject to the choice of initial values of the translation probabilities , which typically are to be taken from subjective human reasoning based upon interpretations of large parallel corpora of natural languages. To avoid this subjectivity, [2] have proposed a choice of initial values based on computation of the standard Mutual Information statistic between the probability distribution (counts) of the parallel corpora under study. Nevertheless , it is obvious that this approach too is heavily dependent on the specific choice of corpora and therefore does not help in obtaining translation probabilities general and independent of specific choices. Here is where there lies novelty in the method proposed in this work, as detailed below.

2. Context and literature survey

Existing literature seeks to find patterns in corpora by trying to impose models of probability distributions especially N Poisson distributions with some success. However, the model fails to capture subjectivity in the degree of relevance of documents to specific terms appearing in the document. [3] [4] and [5] provide examples highlighting this phenomenon.

Also, there is abundance of literature which attribute the tfidf (term frequency , inverse document frequency) model to several corpora as a natural fit to quantitatively model the structure of the documents. The EM algorithm described above may be considered as a natural extension to these models , relying on term counts and providing estimates of the translation probability of the terms. However as already mentioned above these models yield corpora specific quantitative relations and need additional refinement for obtaining the necessary generalization.

3. Uniqueness of Approach in current work

The author re-iterates that the current work is unique to the best of knowledge , in that a new formulation for Mutual Information is proposed and utilized herein to bolster the EM algorithm by supplying it with initial values of translation probabilities which are relatively independent of the corpora instance chosen for the modelling/study.

The classical formulation of the Mutual Information is :

mod_ent[k] = abs(abs(probs[k]+errfact)*log(abs(probs[k]+errfact),2)) - → Formula (1)

This is upgraded in this current work to :

```
mod_ent[k]=abs(abs(probs[k]-
x[minprob]+errfact)*log(abs(probs[k]-
x[minprob]+errfact),2)+abs(probs[k]-
x[maxprob]+errfact)*log(abs(probs[k]-
x[maxprob]+errfact),2)+abs(probs[k]-
x[minchg]+errfact)*log(abs(probs[k]-
x[minchg]+errfact),2)+abs(probs[k]-
x[maxchg]+errfact)*log(abs(probs[k]-
x[maxchg]+errfact),2))
```

→ Formula (2)

where

```
emb = takens(probs)
x=emb[:, 0]
y=emb[:,1]
minprob = np.argmin(x)
maxprob = np.argmax(x)
minchg  = np.argmin(y)
maxchg  = np.argmax(y)
```

errfact=0.0001

.Details of Comparative Experiment performed

The following bi-lingual French-English corpus was utilized to perform the comparative experiment in this work :

@inproceedings{sulem-etal-2015-conceptual,

title = "Conceptual Annotations Preserve Structure Across Translations: A {F}rench-{E}nglish Case Study",

author = "Sulem, Elior and

Abend, Omri and

Rappoport, Ari",

booktitle = "Proc. of S2MT",

month = jul,

year = "2015",

address = "Beijing, China",

publisher = "Association for Computational Linguistics",

url = "https://www.aclweb.org/anthology/W15-3502",

doi = "10.18653/v1/W15-3502",

pages = "11--22",

}

The implementation of the EM algorithm (provided in Appendix I) was tested with the above corpus with comparative initial values using Formulas (1) and (2). The results thus obtained are summarized in the next section.

4. Results Obtained

Following are the respective translation probability (un-normalized) values obtained from application of the EM algorithm with initial values from formulations (1) and (2) :

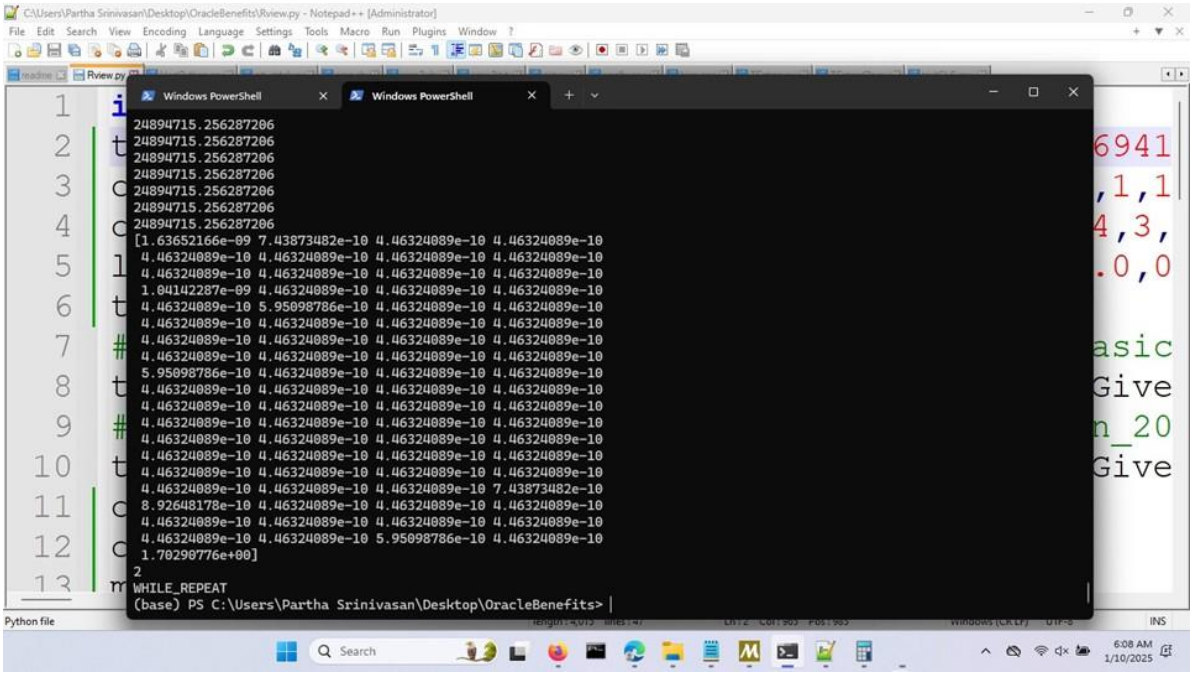

Fig 1 : Translation probabilities with formulation 1

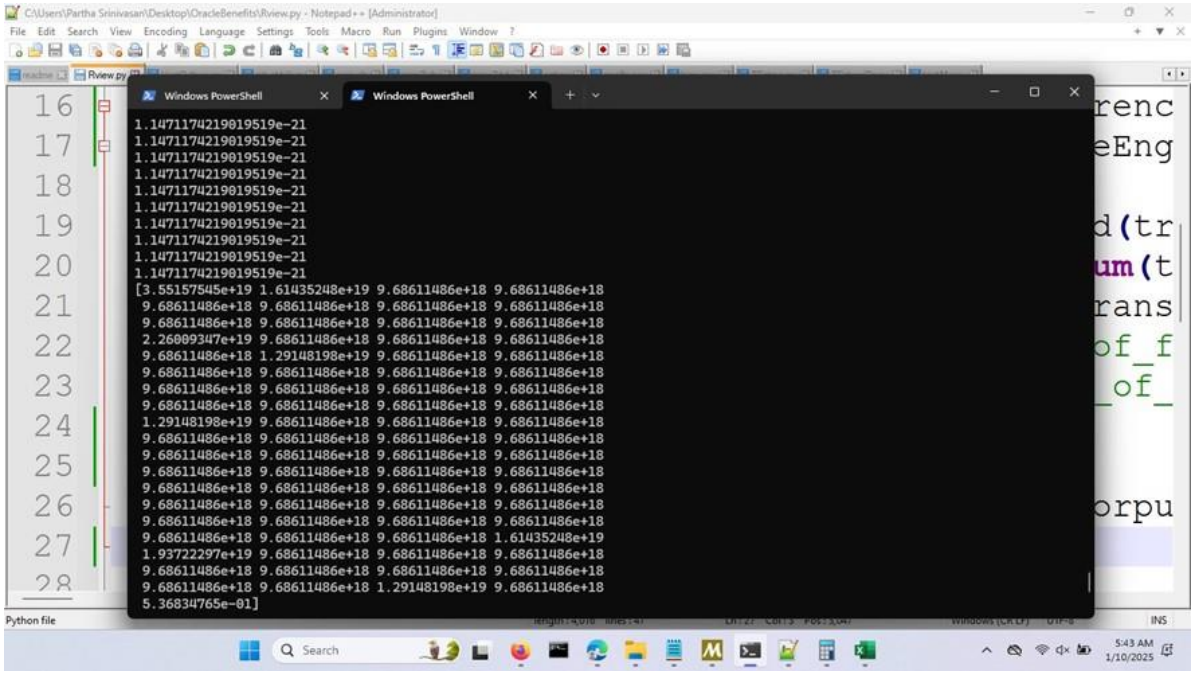

Fig 2 : Translation probabilities with formulation 2

6. Conclusion

Interpretation of the results obtained above shows that when the Mutual Information statistic is defined according to Formulation # 2 , introduced in this work , the translation probabilities are realistic and sensitive to minor semantic variations amongst the terms in the corpus and are free from choices of specific corpus instance.

7. Future Work

It is envisaged that the alternative formulation of Mutual Information found in this work can potentially be used in several contexts and works where this statistic is employed resulting in aggregate refinement in understanding and application of this statistic.

8. REFERENCES

[1] The Mathematics of Statistical Machine Translation: Parameter Estimation Peter E Brown* Vincent J. Della Pietra* Stephen A. Della Pietra* Robert L. Mercer*

[2] Information Retrieval and Information Theory Jaime G. Carbonell ,Daniel Sleator

[3] A language modeling approach to information retrieval J. Ponte, W. Bruce Croft

[4] Term-weighting approaches in automatic text retrieval Gerard Salton, Christopher Buckley

[5] Optimizing Document Indexing and Search Term Weighting Based on Probabilistic Models N. FuhrC. Buckley

Appendix I - Code for EM algorithm

```
import numpy as np

trInitialFrenchGivenEachEnglish =
np.array([0.769410911,0.696303162,1.702907756,1.702907756,1.702907756,0.988253349,1.702907756,1.702907756,0.356411459,1.702907756,0.988253349,0.724751928,0.989933492,0.5845648,0.724751928,1.702907756,1.702907756,0.839857992,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.671605883,1.702907756,1.702907756,1.702907756,1.702907756,0.49652402,1.702907756,1.702907756,1.702907756,0.356411459,1.702907756,1.702907756,1.702907756,0.988253349,1.702907756,1.702907756,1.702907756,0.988253349,0.306665119,1.702907756,0.724751928,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,1.702907756,0.887501652,0.657703452,0.988253349,1.702907756,0.724751928,1.702907756,1.702907756,1.702907756,0.988253349,1.702907756,0.724751928,1.671605883,1.702907756])

cntfinF=np.array([9,3,1,1,1,1,1,1,1,1,1,1,5,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,3,4,1,1,1,1,1,1,1,1,1,2,1])

cnteinE=np.array([29,10,1,1,1,2,1,1,8,1,2,3,10,4,3,1,1,5,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,5,1,1,1,8,1,1,1,2,1,1,1,2,10,1,3,1,1,1,1,1,1,1,1,1,1,1,7,15,2,1,3,1,1,1,2,1,3,2,1])

lambdae=np.array([0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0])

theTargetCount2DArray=np.zeros((77,77))

#trInitialFrenchGivenEachEnglish  =
np.array(<basically_stuff_in_200values_computed_according_to_new_and_old_formulations>)

trDeltaFrenchGivenEachEnglish = trInitialFrenchGivenEachEnglish

#trInitialEnglish =
np.array(<basically_stuff_in_200values_computed_according_to_new_and_old_formulations>)

trDeltaFrenchGivenEachEnglishSave=trDeltaFrenchGivenEachEnglish

corpusSizeFrench=76

corpusSizeEnglish=76

maxtrchange=1

#Repeat{

while(maxtrchange<2) :

 for corpusCounterFrench in range(0,corpusSizeFrench,1) :

  for corpusCounterEnglish in range(0,corpusSizeEnglish,1) :



   transitionProbabilityPairsNumerator = np.prod(trDeltaFrenchGivenEachEnglish)
```

```
   transitionProbabilityPairsDenominator = np.sum(trDeltaFrenchGivenEachEnglish)

   nrbydr=transitionProbabilityPairsNumerator/transitionProbabilityPairsDenominator

   #cntfinF[corpusCounterFrench]=np.sum(<count_of_f_occurrences_In_FrenchCorpus>) #Maybe we can make use of some (correspond) counters

   #cnteinE[corpusCounterEnglish]=np.sum(<count_of_e_occurrences_In_EnglishCorpus>) #Maybe we can make use of some (correspond) counters

   #cntfinF[corpusCounterFrench]=3

   #cnteinE[corpusCounterEnglish]=4

   theTargetCount2DArray[corpusCounterFrench][corpusCounterEnglish] = (np.sum(cntfinF[corpusCounterFrench])+np.sum(cnteinE[corpusCounterEnglish]))*nrbydr

  print(nrbydr)

   #exit()


 for corpusCounterEnglish in range(0,corpusSizeEnglish,1) :

  for corpusCounterFrench in range(0,corpusSizeFrench,1)  :

   #lambdae[corpusCounterEnglish] = np.sum(theTargetCount2DArray[corpusCounterFrench][corpusCounterEnglish])

   lambdae[corpusCounterEnglish] = np.sum(theTargetCount2DArray[corpusCounterFrench])


  for corpusCounterFrench in range(0,corpusSizeFrench,1)  :

   trDeltaFrenchGivenEachEnglish[corpusCounterFrench] = (1/lambdae[corpusCounterEnglish])*(theTargetCount2DArray[corpusCounterFrench][corpusCounterEnglish])/ nrbydr

 trchange=trDeltaFrenchGivenEachEnglish-trDeltaFrenchGivenEachEnglishSave

 trDeltaFrenchGivenEachEnglishSave=trDeltaFrenchGivenEachEnglish

 #maxtrchange=np.argmax(trchange)

 maxtrchange=maxtrchange+1

 print(trDeltaFrenchGivenEachEnglish)

 #trDeltaFrenchGivenEachEnglish.tofile("o_mod",sep=",")

 print(maxtrchange)

 print("WHILE_REPEAT")

#}EndRepeat(ConvergenceCondition)
```

Appendix II

Shell scripts used to obtain count information from the corpus for computing probabilities.

```
for i in {11..324..6}

do

cut -d' ' -f $i passage40.xml | cut -d"" -f 2

done
```

```
for w in `cat together`; do echo $w; done|sort|uniq -c > together_cnt
```