On Establishing

Multilingual Text to Text Similarity

Manikanta Reddy D
Computer Science and Engineering, IIT Kanpur.
manikanta.reddy.d@outlook.com

Shaurya Aarav
Physics, IIT Kanpur.
shauryaaarav@gmail.com

Abstract

In this paper we propose a comprehensive way to learn and map articles of different languages in a semi supervised manner, without using any dictionary based methods. In order to compute text to text similarity our method uses the relative frequency of words, across similar texts, to generate semantic clusters, which are in turn consumed to vectorize texts and study their similarity. Further we will use a LSTM network to learn patterns across languages. Understanding the context and semantics becomes an important feature of of our learning.

Keywords
Natural language processing, Semantic Clusters, LSTM

1 Introduction

Documented knowledge across different languages is easily available to a user in the present era. This large set of text files can be helpful only if there exists an easy way to access them. To improve the accessibility of documents, there is a need of a mechanism to group documents related to each other in terms of their content, irrespective of the language(s) they were originally written.

In our work, we developed a procedure which on receiving a test article, returns a set of labels (from the training data) of multilingual documents, which are contextually similar to it, without employing language translation at any stage. For the purpose of training in our algorithm, we have used the largest text data-set available to us, Wikipedia.

We propose a new method to vectorize the documents by counting the frequency of the words they contain across different semantic word clusters. These language independent word clusters are made with a belief that across contextually similar documents in different languages, words with similar meaning occur with a similar relative frequency.

2 Related Work

Earlier works on clustering multilingual documents involves processes like direct language translation, web searches and common ground representation of language specific word clusters. Evans and Klavans [1] suggested a method in which the entire set of documents was converted into a common language before applying a general monolingual document clustering algorithm. Dani Yogatama [5] created multilingual word clusters using web count as a measure of relatedness among words across different languages. Kiran Kumar N, Santosh GSK, Vasudeva Varma [3] in their paper proposed the vectorization of documents using the "bag of words"" model with enrichment from Wikipedia and representing multilingual clusters on a common ground.

3 Overview of our work

Initially we worked with the classical "word2vec" [4] . We vectorized all the words in our corpus using the above mentioned algorithm and then clustered them using the "K Means Clustering" technique. A vector representation of a document, was then built by using the frequency of each of the K clusters members in that document.

Thus the document was represented as a K dimensional vector, with each component referring to the corresponding frequency of the word cluster. We then performed another K means clustering of the documents in this K dimensional space. These clusters succeeded in grouping similar documents of a single language elegantly, but failed miserably for multilingual documents. The reason, we speculated, is that our document vectors were not language independent as our word clusters themselves were language specific.

In order to achieve language independent word clusters. Using a frequency of a word in different topics, we got a vector which was dependent on topics rather than language, making it language-independent.

We then discuss a neural network based approach for analyzing the texts using LSTM. LSTM will inherently learn the pattern in which text semantics flow and classify them.

4 Data

In order to understand the language model, we had to introduce some amount of supervision. We did this by choosing a list of topics, that will define our model and will be its characteristic. We then extracted documents from Wikipedia on every topic in the list, in languages of our choice, (English, French, Spanish). This decision was made based on the availability of articles.

We here make an important assumption that articles on Wikipedia on a given topic in different languages are semantically coherent, i.e. they deal with the same entity. This is justifiable as wikipedia hosts articles about topics.

After this we strip all the documents of stop-words, to remove common words and then stem every word using an appropriate stemming algorithm for the language.

This will ensure that all documents are clean and contain vocabulary, that capture only the core information.

5 Semantic Similarity

Once all documents are obtained, the first step is to identify semantic relatedness of different words (of all languages). For this we rely on a frequency based approach to map the words into a common, word space, irrespective of their language and perform a clustering algorithm to extract closely related words.

For a given word $w_{\mathcal{L}}$ of language $\mathcal{L}$, let $\tau_{i,\mathcal{L}}$ be the term frequency, normalized for document size, of $w_{\mathcal{L}}$ in the document of topic τ_i in the same language $\mathcal{L}$.

Then the vector representation of $w_{\mathcal{L}}$ will be

$$(\tau_{1,\mathcal{L}}(w_{\mathcal{L}}), \tau_{2,\mathcal{L}}(w_{\mathcal{L}}), \ldots, \tau_{\mathcal{T},\mathcal{L}}(w_{\mathcal{L}}))$$

Where $\mathcal{T}$ is the number of topics chosen. Same procedure is used for each word in their respective language. Thus, all words are mapped into the same space of dimension $\mathcal{T}$, irrespective of their language.

We refer to this $\mathcal{T}$ dimensional space as *Word Space* as every point in this space represents a word. We now make a hypothesis that, in this word space, if two words are semantically similar they exist in close neighborhoods of each other.

We support our hypothesis with the following example. The relative number of times the word *physics* would appear in an article on Physics in English would be approximately same to the number of times *physique* would appear in an article on Physics in French. Hence, we can generate vectors of all words in the same space, where words that represent the same entity should lie close to each other.

We can extract information about the semantics of the data by finding very closely packed clusters in the word space. These clusters can be found by the regular K-Means algorithm. Using the label of the cluster a word belongs to, we can associate with each word a language independent **meaning**.

We provide an example to further justify our assumption. We stemmed two documents on Pokemon, one in English and the other in Spanish. **Table1** and **Table2** show the relative frequency of top six words occurring in both documents after being stemmed. These distributions are strikingly similar. *pok* matches with *pok*, *game* matches with *jueg* and so on.

Table 1: Pokemon English

Order	Filtered word count	Occurrences	Percentage
1	pok	302	13
2	mon	285	12
3	game	116	5
4	release	56	2
5	player	35	1

Table 2: Pokemon Spanish

Order	Unfiltered word count	Occurrences	Percentage
1	pok	191	13
2	mon	179	12
3	jueg	48	3
4	videojueg	24	2
5	nintend	23	2

Table 3: Pokemon Spanish

Some top phrases containing 2 words (without punctuation marks)	Occurrences
de pokmon	41
de la	36
en el	30
de los	22
en la	20
a la	16
un pokmon	15
la serie	13

Table 4: Pokemon English

Some top phrases containing 2 words (without punctuation marks)	Occurrences
of the	69
of pokmon	42
the pokmon	41
in the	39
to the	25
with the	20
and the	20
a pokmon	16

Looking deeper into the evidence if we consider occurrences of pairs of words in the documents, even they have a striking resemblance in both the languages. **Table 3** and **Table 4** demonstrate this.

6 Text Representation

Now that we have a language independent model for words, we will use it to represent text of any language in a unified manner. This again is a frequency based approach and we will generate something that we call ***bag-of-meanings***.

Any given document of any language d is stripped of stop-words and its words are stemmed appropriately. Let $\mathcal{T}_{ws}(meaning)$ be the normalized count of how many times words belonging to the semantic cluster *meaning* appear in d .

Then the vector representation of d will be

$$((\tau_w s(meaning1), \tau_w s(meaning2), \ldots, \tau_w s(meaningS))$$

Where S is the number of semantic clusters and call this S dimensional space a *Topic Space* as every point in this space defines a unique Topic.

This S dimensional vector representation of a piece of text is independent of its language and represents the bag-of-meanings it holds within it.

We make yet another hypothesis that, if two pieces of texts have similar semantic histograms then they are probably discussing the same thing.

So we consider the semantic histogram of any article as its descriptor, it can viewed as its language independent vector. We can now project all the documents into this hyperspace of the semantics and study their behavior.

6.1 Score Function

We defined a scoring function as follows:

$$score(y) = \sum_{\mathcal{T}} (\frac{x(z)}{Distance(y,z)})^2$$

Where $x(z) \in (1, N)$, stores the index of the contextually similar document z (to the test case) in an array arranged in ascending order of the distance between matching documents in the topic space with $\mathcal{T}$ best topic matches. $Distance(y, z)$ is the distance between the test document and the correctly matched training document in the topic space.
We normalize the above score with square sum of the distance between the test document and all the documents matched up to $\mathcal{T}$ = number of languages.
The scoring function essentially rewards a match based on its rank and penalizes it for its distance from query.

6.2 Results

6.2.1 English + French

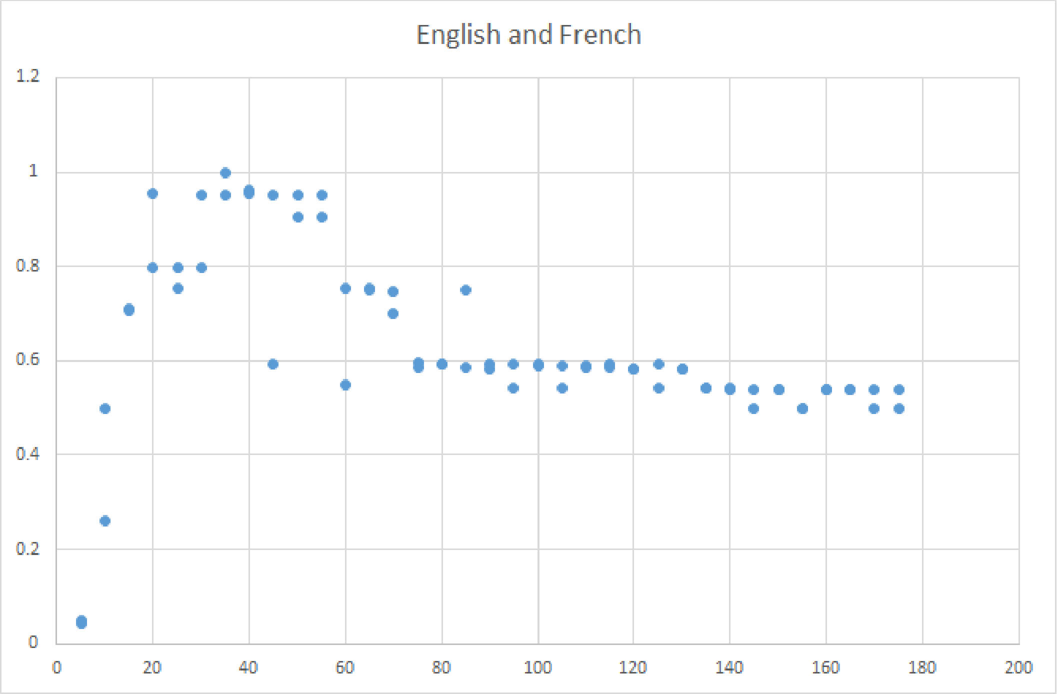

Figure 1: Semantic Clusters vs Score

Topic	Neighbour1	Neighbor2	Neighbor3	Neighbor4	Score
India test en	Politics	India	India	Politics	0.16
Pokemon test en	pokemon	pokemon	Politics	Energy	1
India test fr	Energy	Politics	pokemon	Politics	0
Pokemon test fr	pokemon	pokemon	Energy	Politics	1

Table 5: A sample of our results with two languages

We notice how the variation of number of semantic clusters affects the score. Large number of clusters leads to sparse distribution of meanings and decrease the score. Less number of clusters just means that there are not just enough meanings to represent the text.

6.2.2 English + French + Spanish

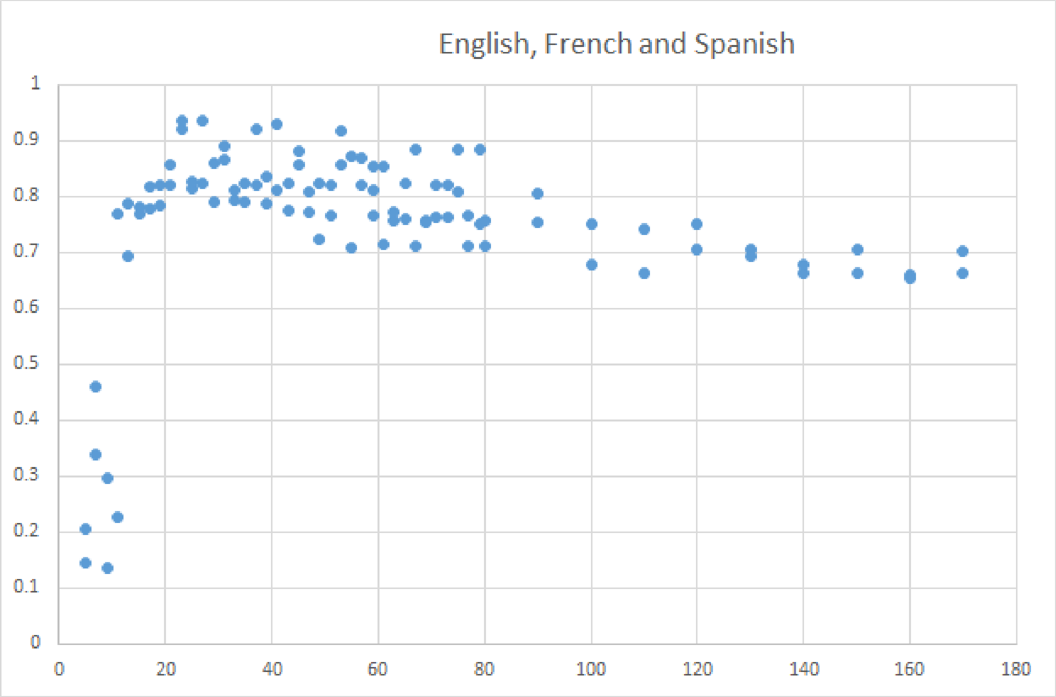

Figure 2: Semantic Clusters vs Score

Topic	Neighbour1	Neighbor2	Neighbor3	Neighbor4	Score
India test en	India	India	Politics	Beatles	0.94
Pokemon test en	pokemon	pokemon	pokemon	Beatles	1
India test fr	India	Politics	Beatles	India	0.66
Pokemon test fr	pokemon	pokemon	Beatles	pokemon	0.97
India test es	India	India	Beatles	India	0.95
Pokemon test es	pokemon	pokemon	pokemon	Beatles	1

Table 6: A sample of our results with three languages

India_test_en contains an article about a country which is similar (in terms of context) to an article on *Politics* but dissimilar to one on *Pokemon*, so the neighbor of *India_test_en* should be rewarded, but our score measure does not incorporate relatedness between topics. If such a test measure is built, we can tune the parameters to generate better results.

Pokemon_test shows perfect results as the training set had no other article on a fictional animated series. So this document matched exactly to its counterparts as all other topics were very different contextually.

7 Long Short Term Memory

The previous bag-of-meanings model does not exactly capture the context of the document. This is a direct limitation of using histogram based methods. They are able to capture the global status of the problem but are incapable of identifying the pattern within.
We will be using LSTM [2] to learn the context/ pattern in writing articles on a Topic.

An LSTM is a Recurrent architecture, well suited to learn from experience to classify events. A typical LSTM block has been shown below.

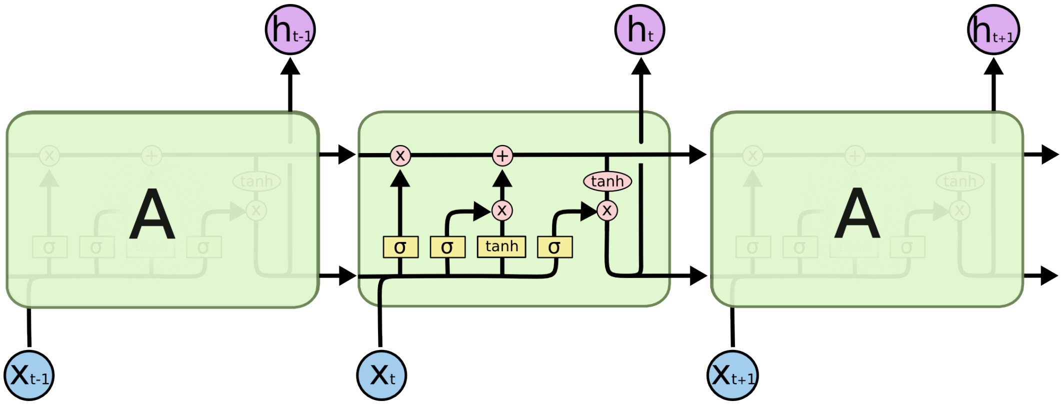

Figure 3: A simple LSTM block with one input and output gate.
Source: colah

In order to utilize LSTM, we first represented text in a language independent fashion, by replacing every word in the text by its *meaning*. Hence the text will now be a continuous sequence of *meanings*. This sequence can now be learned by a LSTM network to recognize the pattern in writing the text.

7.1 Results

Topic	Precision	Recall	F1 Score	Support
Pakistan	1	1	1	1231
The_Beatles	0.98	1	0.99	1504
Chemistry	1	1	1	1091
India	1	1	1	1336
Energy	1	1	1	553
Tennis	1	0.96	0.98	1129
Wikipedia	1	1	1	1447
English_language	0.95	1	0.98	753
Adele	1	1	1	690
French_language	1	0.95	0.98	827
Politics	1	1	1	853
Baboon	1	1	1	316
pokemon	1	1	1	764
Average	0.99	0.99	0.99	12494

Table 7: Performance of LSTM

It is evident that this method is extremely efficient in classifying text into their basic topics. The training set contained sentences of different topics in different languages.

This suggests that the sentences that make up an article on a given topic follow a similar pattern in all languages (At least they do in English, French and Spanish). We can now use the probability votes given by every sentence to which class they belong, to generate an aggregate probability of the collection of sentences.

7.2 LSTM Architecture

We built our neural network in the keras framework running on theano.

The LSTM architecture we built for this purpose consists of an Embedding layer that runs the Word2Vec algorithm to embed the semantic labels in the text into a space of 64 dimensions.

Word2Vec works here, as all the documents are now in a common language of semantics. And we have already seen how well word2vec works for mono lingual documents.

The second layer consists of a LSTM block initialized by Glorot-Style uniform weights, with inner weights being initialized in an orthogonal fashion.

The activation function of the neural net is a hard sigmoid function.

LSTM networks are prone to overfitting. So we use a dropout of 0.5 to reduce it.

With a batch size of 256 sentences, every epoch of this network takes about 8s to run on a GPU. This method is computationally expensive during the training phase. But the testing is executes in almost no time. We split all the sentences in our corpus into 25% validation nd a 75% training set.

8 Conclusion

We discussed our approach to compute semantic relatedness between texts in a language independent manner. Our algorithm is extremely fast in execution phase and is near unsupervised during training. Once trained the computational complexity of a single query with a piece of text of length (x) is

$$O(L * constant)$$

Where $constant = KT$, K is the number of semantic clusters and T is the number of training topics, which are fixed for a given trained model. So the time complexity of the algorithm is Linear.

References

[1] David Kirk Evans and Judith L Klavans. A platform for multilingual news summarization. 2003.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] Kiran Kumar, GSK Santosh, and Vasudeva Varma. *Multilingual document clustering using wikipedia as external knowledge*. Springer, 2011.

[4] T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

[5] Dani Yogatama. Clustering multilingual documents by estimating text-to-text semantic relatedness. 2010.