Polyglot Word Embeddings Discover Language Clusters

2020-02-03
machine-learning (/tags/machine-learning.html), ai (/tags/ai.html), ml (/tags/ml.html), nlp (/tags/nlp.html), language (/tags/language.html), linguistics (/tags/linguistics.html), natural-language-processing (/tags/natural-language-processing.html), nlp (/tags/nlp.html)

Polyglot word embeddings obtained by training a skipgram model on a multi-lingual corpus discover extremely high-quality language clusters.

These can be trivially retrieved using an algorithm like $k-$Means giving us a fully unsupervised language identification system.

Experiments show that these clusters are on-par with results produced by popular open source and commercial models.

We have successfully used this technique in many situations involving several low-resource languages that are poorly supported by popular open source models.

This blog post covers methods, intuition, and links to an implementation based on 100-dimensional FastText embeddings.

Background

The skipgram model takes a word as input and predicts its context.

The pipeline maps a one-hot representation of a word in the vocabulary *V* to a dense, real-valued vector called a *word embedding*.

The training scheme for this model involves positive examples obtained from words and their actual contexts in the corpus, and negative examples obtained using *negative sampling* - where words that don't appear in the desired word's contexts are sampled.

Further, fastText introduces a sub-word model where a word's representation is the sum of the representations of the constituent *n*-grams.

This sub-word information produces consistent representations for words despite spelling variations or errors - a useful feature when working with noisy social media text.

We'll henceforth concern ourselves with 100-dimensional FastText (https://fasttext.cc/) word embeddings and refer to them as `FastText-100`. These results are consistent with the 300-dimensional variant as well.

A *document embedding* is a single vector for a full document (sentence, paragraph, social media post etc.).

There are several popular ways of obtaining these but we'll just use the FastText default:

- obtain the word-vectors for all words in the document,
- normalize them, and
- average them to obtain a single 100-dimensional vector for the whole document.

Polyglot Embeddings

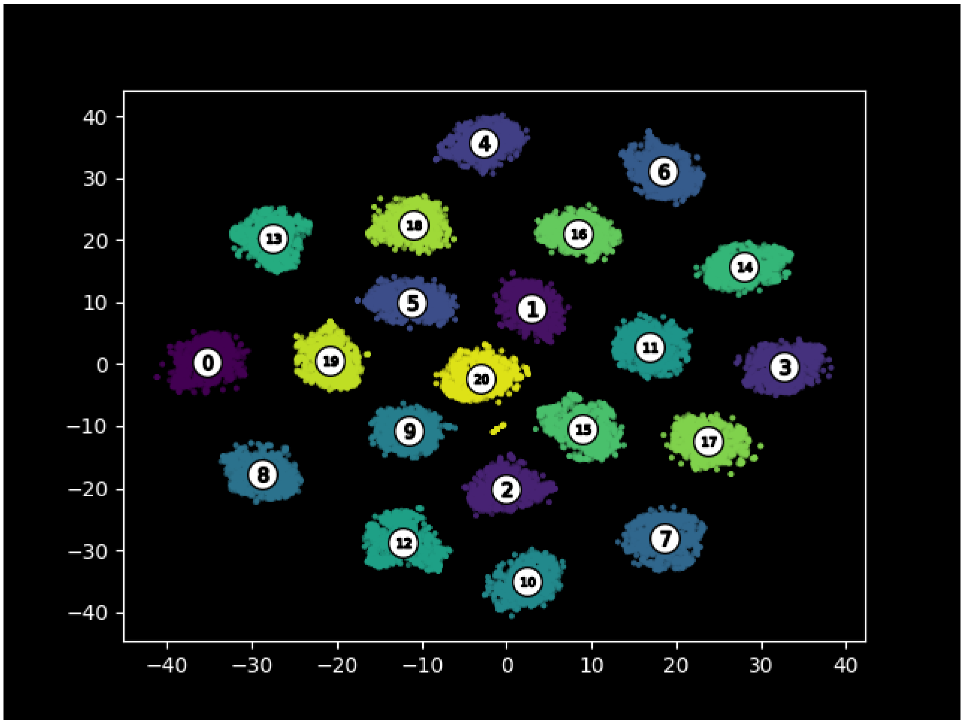
I am going to use a pre-processed Europarl corpus where:

- the text is lowercased and punctuation stripped.
- each line corresponds to a distinct *document*, and
- the full corpus is shuffled.

The corpus contains **21** languages. The documents are shuffled and `FastText-100` embeddings trained on this 21-corpus set.

You can download it (and models and everything) here (http://shriphani.com/europarl.zip)

We can visualize the document embedding space using a TSNE plot:



We observe 21 clear, well defined clusters.

Manual inspection shows that there is exactly **one cluster per language** in the corpus.

A simple clustering algorithm like $k-$Means is able to retrieve these clusters and on inspection we notice that the corpus is split into 21 parts - each containing documents written in exactly one language.

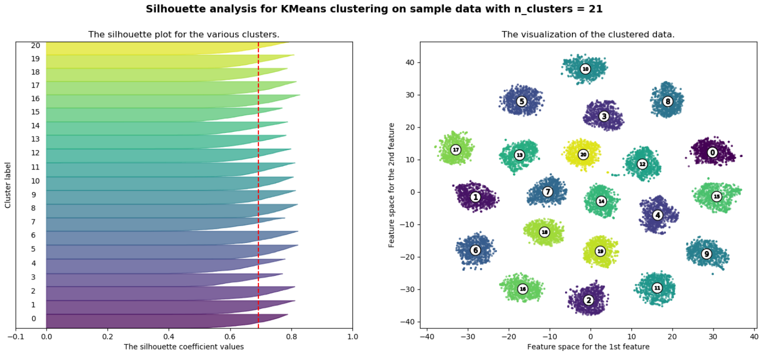
A Language Identification System

Following the steps above, for a test document a cluster membership test is good enough.

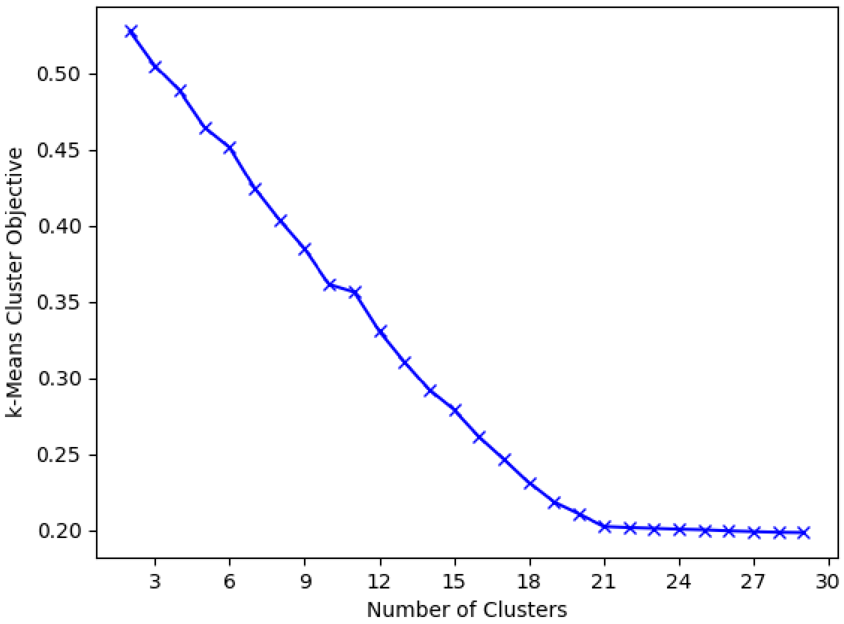
What remains to be discussed is picking a good *k* value for the *k*-Means algorithm.

In our experience, the popular silhouette heuristic (https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html) or the elbow method (https://en.wikipedia.org/wiki/Elbow_method_(clustering)) are quite capable of picking this value.

For the Europarl corpus above, here is the silhouettes plot:



And here is the elbow plot:



In both cases, we see 21 as the optimal number of clusters for this corpus.

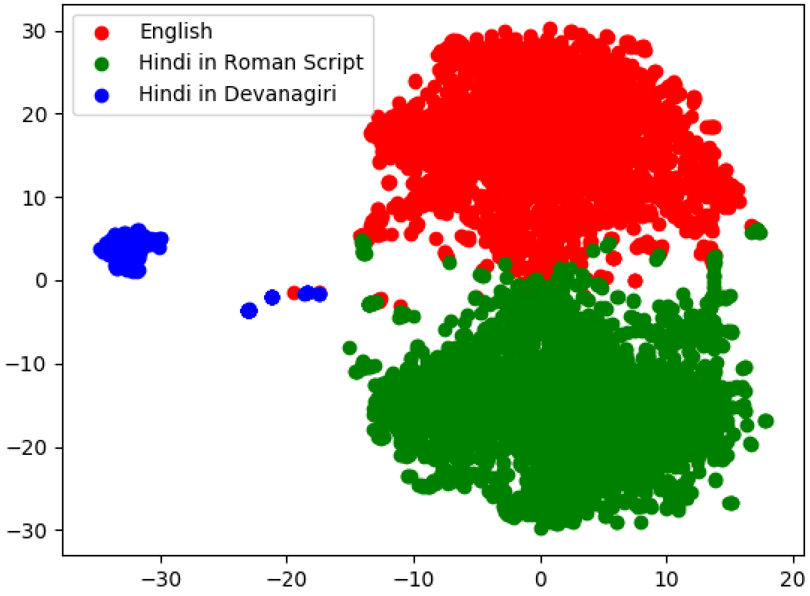
How Good Are These Clusters?

We run human evaluations on several corpora and report performance against popular open and commercial models. These corpora contain several low-resource languages and thus popular language identification systems are unable to identify them:

- $\mathcal{D}_{IndPak}$ - Indian and Pakistani News Channels - Contains English, Romanized and Devanagiri Hindi.
- $\mathcal{D}_{ABP}$ - Bengali News Network - English, Romanized Hindi, Romanized and native script Bengali.
- $\mathcal{D}_{OTV}$ - Oriya News Network - English, Romanized Hindi, Romanized and native script Oriya.

In all these cases, our model is able to perform **exceptionally well**. In cases where it makes sense, performance is on par with popular and commercial models. For full details see our paper (https://arxiv.org/pdf/1909.12940.pdf).

I'm attached the embedding space from $\mathcal{D}_{IndPak}$ here:



Notice how clean the full embedding space is.

Intuition

Why is this happening?

The skipgram training scheme predicts a context for an input word.

A Hindi word's context is likely to consist of other Hindi words; and examples retrieved in the negative sampling phase contain different languages (Hindi words are unlikely to exist in an English context).

Similarly, an English word is likely to occur in a context consisting of other English words.

It just so happens that the skipgram training scheme is perfectly designed to separate out a multilingual corpus into its monolingual components.

Note that inside the embedding subspace corresponding to a language, we observe similar properties typically observed when these word embeddings are trained on monolingual corpora.

i.e. standard stuff like analogies, semantic similarities work very well.

In The Wild

We have used this technique in a variety of text analyses centered in South and South-East Asia.

In particular, South Asian social media users used Romanized variants of their native languages that are mostly unsupported by popular language identification systems.

This unsupervised method is highly capable in such situations eliminating significant annotation requirements.

In this paper analyzing the Rohingya crisis, from among dozens of languages (many low-resource), this technique was utilized to extract out English social media posts.

In an analysis of the India-Pakistan crisis of 2019, this technique was used to separate out Romanized Hindi and English for further analysis, and in soon to be published work analyzing the 2019 Indian election, nearly a dozen Indian regional languages were extracted and analyzed with zero annotation burden. In almost all these cases existing solutions performed poorly or involved prohibitive costs.

For linguistically diverse regions, we foresee that polyglot embeddings are going to be an important NLP pipeline component.

In Related News

Unsupervised or low-supervision methods have been increasingly deployed in multlingual settings.

For instance, embeddings learned by machine translation models seem to be organized along linguistic lines (https://arxiv.org/abs/1909.02197).

A clever training setup was used for unsupervised machine translation in this paper (https://arxiv.org/abs/1804.07755).

Links

- Github: link (https://github.com/shriphani/polyglot-toolbox)
- Paper: arXiV (https://arxiv.org/abs/1909.12940) | PDF (https://arxiv.org/pdf/1909.12940.pdf)

Cite

```
@inproceedings{kashmir,
  title={Hope Speech Detection: A Computational Analysis of the Voice
of Peace},
  author={Palakodety, Shriphani and KhudaBukhsh, Ashiqur R. and Carbo
nell, Jaime G},
  booktitle={Proceedings of ECAI 2020},
  pages={To appear},
  year={2020}
}
```

Comments **Community** Ava

♡ **Recommend** Tweet Share

Sort by Best

Start the discussion…

Be the first to comment.

2019 Sculpture Portfolio → (/2020/01/29/2019-sculpture-portfolio/)

Twitter: @shriphani (https://twitter.com/shriphani)
Instagram: @life_of_ess
(https://www.instagram.com/life_of_ess/)
Fortior Per Mentem