Vector of Locally Aggregated Embeddings for Text Representation

Hadi Amiri[a] and Mitra Mohtarami[b]

[a]Harvard, Department of Biomedical Informatics, Boston, MA, USA

[b]MIT, Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA

hadi@hms.harvard.edu, mitra@csail.mit.edu

Objectives

We study the effect of information loss associated with average word embeddings and develop algorithms that are robust against information loss for text representation. We show that divergence of word embeddings from their corresponding average can be considered as a good proxy to quantify information loss.

Quantifying Information Loss

Let's assume a d-dimensional word embedding space. We quantify the amount of information loss in the average word embedding vector of a given document $\mathbf{S} \in \mathbb{R}^{n \times d}$ by computing the average divergence (or distance) between its word embeddings, $\mathbf{w}_i \in \mathbb{R}^d \; \forall i \in \{1 \dots n\}$, and their average vector, $\overline{\mathbf{s}} = 1/n \sum_i \mathbf{w}_i$, $\overline{\mathbf{s}} \in \mathbb{R}^d$, as follows:

$$divergence = \frac{1}{n} \sum_i (1 - cosine(\overline{\mathbf{s}}, \mathbf{w}_i)). \quad (1)$$

Figures 2 shows strong positive correlation between divergence and document length across long and short text datasets. Thus we use divergence from mean as a good proxy to quantify information loss associated with average embeddings.

Model Overview

We present *Vector of Locally Aggregated Embeddings* (`VLAEs`) for effective and, ultimately, lossless representation of textual content. Our model encodes each input text by identifying and integrating the representations of its semantically-relevant parts. The proposed model consists of a clustering component according to which semantically-relevant parts of each input are identified, and an autoencoding component that integrates representations of these parts. Our model improves the classification performance of current state-of-the-art deep averaging networks across several text classification tasks.

Vector of Locally Aggregated Embeddings (VLAE)

Clustering and Encoding

As Figure 1 shows, we first cluster the embedding space into k clusters over a global vocabulary $\mathcal{V}$. We then compute cluster-level representations for each given document $\mathbf{S} \in \mathbb{R}^{n \times d}$ by averaging its word embeddings at each cluster. We can then represent a document by concatenating these cluster-level representations, $\mathbf{A} \in \mathbb{R}^{d \times k}$. $\mathbf{A}$s have fixed length and can be readily used as features, but they have large size of $(d \times k)$. We tackle this issue by autoencoding $\mathbf{A}$ into vector $\mathbf{a} \in \mathbb{R}^{d \times m}$ where $m < k$ is the dimensionality reduction parameter. Our intuition is that if $\mathbf{a}$ leads to a good reconstruction of $\mathbf{A}$, it has retained all information available in the input.

Illustration of VLAEs

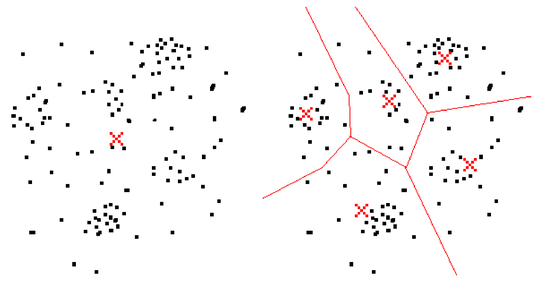

Figure 1: Squares show word embeddings of a document and crosses show their average. (a): the case of high information loss or divergence from mean, (b): shows average word embeddings at cluster level which are used to produce `VLAE`s.

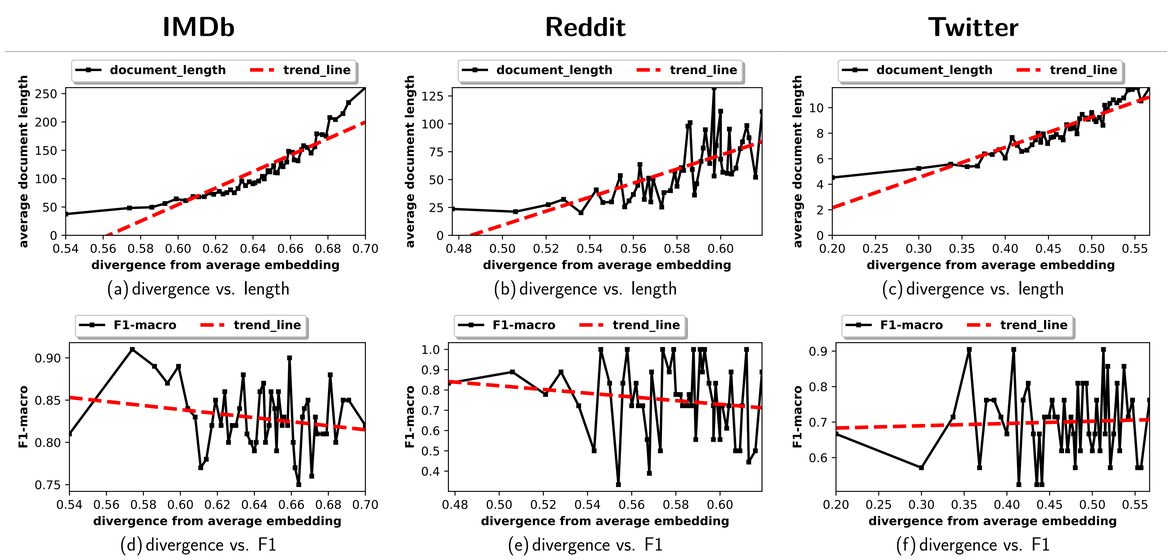

Figure 2: Quantification of information loss associated with average word embeddings. Divergence indicates the average distance between individual word embeddings (of size $d = 300$) and their average embedding, see Equation (1). (a-c): show strong positive correlation between divergence and average document length (#words) across datasets. (d-f): show macro F1 performance of a deep averaging network across datasets: performance considerably drops for higher values of information loss/divergence, e.g. divergence values above 0.55. Note that, we sort and bin instances based on their divergence values and report average length (a-c) and macro-F1 (d-f) for each bin.

Experiments

Tasks

- Sentiment classification (IMDb)
- Disease-text classification (Reddit)
- Churn prediction (Twitter)

Settings

- We use 300-dimensional word embeddings ($d = 300$) provided by Google.
- We train `word2vec` on unlabeled data for d>300.
- We set the dimensionality reduction parameter m from $\{1 \dots 4\}$ using validation data.
- We set the number of clusters k for VLAEs by choosing the optimal k from $\{2^i\}_{i=1}^{7}$ using validation data. We learn optimal k with respect to task, but not embedding space, due to significant density of embedding space.
- We don't update word embeddings during training to directly evaluate the averaging effect.

Results

`Avg_small` and `Avg_large` represent input documents by average word embedding of size $d = 300$ and $d = m \times (300 + k)$ respectively. `Avg_large` has the exact same parameter size as our model (`VLAE`).

	`Avg_small`	`Avg_large`	`VLAE`
IMDb	83.11	78.52	**85.72***
Reddit	59.42	62.72	**66.10***
Twitter	61.42	**73.08***	72.62
AVG	67.98	71.44	**74.81**

Table 1: Macro-F1 performance across datasets.

Takeaways

1. Significant information loss can occur when word embeddings are averaged, in particular, when representing longer documents.
2. Information loss can inversely affect classification performance on longer texts.