A Study of Methods for the Generation of Domain-Aware Word Embeddings

Dominic Seyler (dseyler2@Illinois.edu)

ChengXiang Zhai (czhai@Illinois.edu)

University of Illinois at Urbana-Champaign

Motivation

- “Out-of-the-box” word embeddings are trained on large-scale general-purpose corpora
- Non-specific for application domain: Often perform poorly on specialized domains
- Training difficult: Application domains often have small corpora, which yield low quality embeddings
- **Research question:** How to best leverage general corpora (broad vocabulary but flat domain coverage) and domain corpora (narrow vocabulary but deep domain coverage)?

How to combine general and domain information?

Combination can be done at the corpus level, model level and vector level.

As different models have been heavily studied, we focus on model-independent solutions

We use word2vec as our model, but any word embedding model can be used

We propose two vector and one corpus-level method

Methodology

01

Interpolation of Aligned Embeddings

02

Concatenation of Embeddings

03

Weighted Fusion of Training Data

Interpolation of Aligned Embeddings
(INTERPOL)

- Idea: <u>Weighted addition of the word vector spaces</u>

$$E^{smooth} = (1 - \lambda)E^{domain} + \lambda E^{general}$$

- E^{smooth}= Smoothed word embedding vector space
- E^{domain}= Domain word embedding vector space
- $E^{general}$= General word embedding vector space
- λ = weighting parameter (controls for amount of "smoothing" using the general embedding)

Interpolation of Aligned Embeddings
(INTERPOL)

- Vectors need to be transformed before addition, such that

$$E^{general}W = E^{domain}$$

- W is a transformation matrix that can be found using stochastic gradient decent

$$\min_W \sum_i ||WE_i^{general} - E_i^{domain}||^2$$

- Incorporate W in previous equation

$$E^{smooth} = (1-\lambda)E^{domain} + \lambda WE^{general}$$

Concatenation of Embeddings
(CONCAT)

- Concatenate word embeddings of E^{domain} and $E^{general}$ into a single vector E^{concat}

$$E^{concat} = E^{domain} || E^{general}$$

- Intuition: model will learn to prioritize certain embedding dimensions using the training data.

Weighted Fusion of Training Data
(FUSION)

- Combine general and domain corpora in a principled manner before training embeddings
- Control for amount of domain data with N ("domain duplication factor")
- N specifies the number of duplications of the domain data
- Two extreme scenarios: No domain data ($N = 0$), only domain data ($N \to \infty$)

Experiments

Experimental Setup

- Dataset
  - MalwareTextDB [Lim(2017)] **:** classifying relevant sentences for inferring malware actions and capabilities (binary sentence classification).
  - Randomly sample the dataset into training (80%, 10,334 sentences), development (10%, 1,292 sentences) and testing (10%, 1,292 sentences).
  - Performance on dataset measured using F_1 score.
- Embedding Algorithm
  - word2vec [Mikolov(2013)]
- Classification framework:
  - CNN: Convolutional Neural Network model from [Kim(2014)].

[Lim(2017)] Lim, Swee Kiat, et al. "Malwaretextdb: A database for annotated malware articles." *ACL*. 2017.
[Mikolov(2013)] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *NIPS*. 2013.
[Kim(2014)] Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." *EMNLP*. 2014.

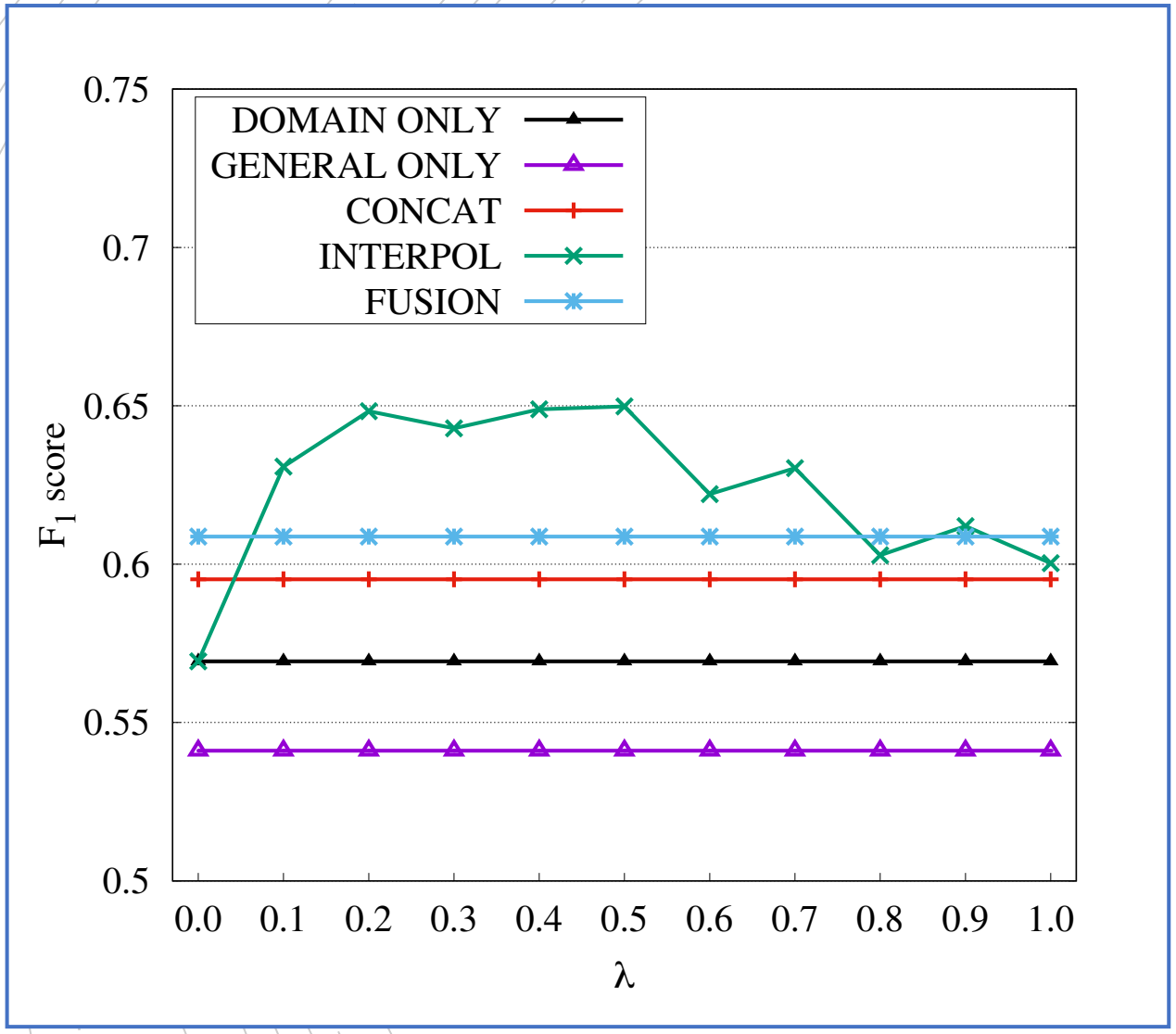
Experimental Setup

- Baselines
  - GENERAL: embeddings trained on a large general-purpose corpus (Wikipedia).
  - DOMAIN: embeddings trained on a small domain corpus. We train cybersecurity embeddings from a crawl of security-related webpages.
- Hyperparameters:
  - Weighting parameter λ: range $[0,1]$ with increments of 0.1
  - Domain Duplication factor N : No domain data ($N = 0$) to dominated by domain data ($N = 100$)
  - Embedding dimensions dim: 100, 200, 300

dim	Dev	Test	Model
300	0.5693	0.5209	DOMAIN
	0.5411	0.4773	GENERAL
	0.5952	0.5331	CONCAT
	0.6087	0.5429	FUSION (N=15)
	0.6498	**0.5778**	INTERPOL ($\lambda = 0.5$)
200	0.6340	0.6013	DOMAIN
	0.5411	0.5168	GENERAL
	0.5912	0.5164	CONCAT
	0.6131	**0.6069**	FUSION (N=50)
	0.6655	0.6053	INTERPOL ($\lambda = 0.2$)
100	0.5934	0.5685	DOMAIN
	0.5271	0.5090	GENERAL
	0.5688	0.5220	CONCAT
	0.6156	0.5607	FUSION (N=100)
	0.6446	**0.6239**	INTERPOL ($\lambda = 0.4$)

Comparison of Models

- INTERPOL performs best on both datasets overall
- CONCAT it is often outperformed by the DOMAIN baseline.
- FUSION outperforms other baselines under certain conditions
- DOMAIN performs better than GENERAL in majority of cases
- combining domain and general data is generally beneficial



Lambda Parameter

- Small amounts of general information can push the performance past other models.
- Values higher than 0.5 hurt performance.
- Transforming the general embeddings with the transformation matrix boosts performance

DOMAIN		GENERAL		INTERPOL	
flaw	.795	bugs	.817	flaw	.758
glitch	.732	worm	.711	vulnerability	.690
vulnerability	.719	stagefright	.702	issue	.681
issue	.686	mouse	.679	bugs	.660
bugs	.644	flappy	.670	glitch	.629
defect	.624	beetle	.654	problem	.596
loophole	.611	blob	.651	defect	.550
problem	.610	gizmo	.641	flaws	.535
weakness	.571	stink	.637	stagefright	.524
flaws	.568	critter	.633	loophole	.519

Qualitative Evaluation

- “bug” is ambiguous in different domains
- DOMAIN lists mostly cybersecurity related terms
- GENERAL captures mostly the biological meaning of bug
- INTERPOL method can introduce “stagefright” to the most similar words and more general terms are ranked higher.

Conclusions

- Generally, combining domain and general data is beneficial
- Interpolation performs best and should be preferred, because of its flexibility using λ.
- Concatenation it is often outperformed by the baselines
- Fusion model can achieve good performance, but costly to train
- Consistently find that general embeddings are the least effective in almost all settings.
- Qualitative analysis indicates meaningful transformation of the vector space when interpolation method is used

A Study of Methods for the Generation of Domain-Aware Word Embeddings

Dominic Seyler (dseyler2@Illinois.edu)

ChengXiang Zhai (czhai@Illinois.edu)

University of Illinois at Urbana-Champaign

ILLINOIS

Thank You!

more info at: https://dominicseyler.com