Dynamic Bernoulli Embeddings for Language Evolution

Maja Rudolph, David Blei

Columbia University, New York, USA

Abstract

Word embeddings are a powerful approach for unsupervised analysis of language. Recently, Rudolph et al. (2016) developed exponential family embeddings, which cast word embeddings in a probabilistic framework. Here, we develop *dynamic embeddings*, building on exponential family embeddings to capture how the meanings of words change over time. We use dynamic embeddings to analyze three large collections of historical texts: the U.S. Senate speeches from 1858 to 2009, the history of computer science ACM abstracts from 1951 to 2014, and machine learning papers on the Arxiv from 2007 to 2015. We find dynamic embeddings provide better fits than classical embeddings and capture interesting patterns about how language changes.

1. Introduction

Word embeddings are a collection of unsupervised learning methods for capturing latent semantic structure in language. Embedding methods analyze text data, learning distributed representations of the vocabulary to capture its cooccurrence statistics. These learned representations are then useful for reasoning about word usage and meaning (Harris, 1954; Rumelhart et al., 1986). With large data sets and approaches from neural networks, word embeddings have become an important tool for analyzing language (Bengio et al., 2003; Mikolov et al., 2013c;b;a; Pennington et al., 2014; Levy & Goldberg, 2014; Arora et al., 2015).

Recently, Rudolph et al. (2016) developed *exponential family embeddings*. Exponential family embeddings distill the key assumptions of an embedding problem, generalize them to many types of data, and cast the distributed representations as latent variables in a probabilistic model. They encompass many existing methods for embeddings and open the door to bringing expressive probabilistic modeling (Bishop, 2006; Murphy, 2012) to the problem of learning distributed representations (Bengio et al., 2003).

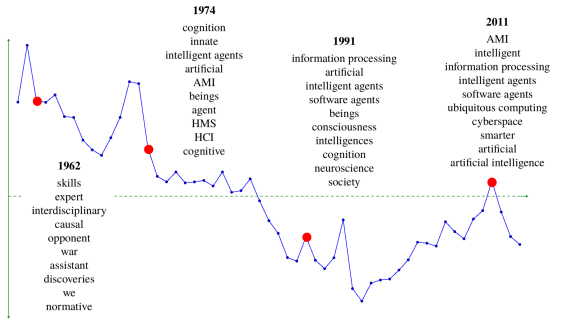
Here we use exponential family embeddings to develop *dynamic word embeddings*, a method for learning distributed representations that change over time. Dynamic embeddings analyze long-running texts, e.g., documents that span many years, where the way words are used changes over time. The goal of dynamic embeddings is to characterize and understand those changes.

Figure 1 illustrates the approach. It shows the changing representation of INTELLIGENCE in two corpora, the collection of computer science abstracts from the ACM 1951–2014 and the U.S. Senate speeches 1858–2009. On the y-axis is "meaning," a proxy for the dynamic representation of the word; in both corpora, its representation changes dramatically over the years. To understand where it is located, the plots also show similar words (according to their changing representations) at various points. Loosely, in the ACM corpus INTELLIGENCE changes from government intelligence to cognitive intelligence to artificial intelligence; in the Congressional record INTELLIGENCE changes from psychological intelligence to government intelligence. Section 3 gives other examples from these corpora, such as IRAQ, DATA, and COMPUTER.

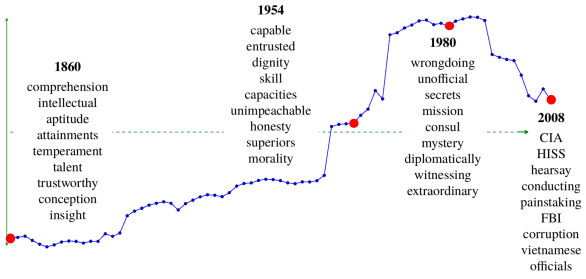
In more detail, a word embedding uses representation vectors to parameterize the conditional probabilities of words in the context of other words. Dynamic embeddings divide the documents into time slices, e.g., one per year, and cast the embedding vector as a latent variable that drifts via a Gaussian random walk. When fit to data, the dynamic embeddings capture how the representation of each word drifts from slice to slice.

Section 2 describes dynamic embeddings and how to fit them. Section 3 studies this approach on three datasets: 9 years of Arxiv machine learning papers (2007–2015), 64 years of computer science abstracts (1951–2014), and 151 years of U.S. Senate speeches (1858–2009). Dynamic embeddings give better predictive performance than existing approaches and provide an interesting exploratory window into how language changes.

Related work. Language is known to evolve (Aitchison, 2001; Kirby et al., 2007) and there have been several lines of research around capturing semantic shifts. Mihalcea & Nastase (2012); Tang et al. (2016) detect semantic changes of words using features such as part-of-speech tags and en-



(a) INTELLIGENCE in ACM abstracts (1951–2014)



(b) INTELLIGENCE in U.S. Senate speeches (1858–2009)

Figure 1. The dynamic embedding of INTELLIGENCE reveals how the term's usage changes over the years. The y-axis is "meaning," a one dimensional projection of the embedding vectors. For selected years, we list words with similar dynamic embeddings.

tropy and Sagi et al. (2011); Basile et al. (2014) employ latent semantic analysis and temporal semantic indexing for quantifying changes in meaning.

Most closely related to our work are methods for dynamic word embeddings (Kim et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016). These methods train a separate embedding model for each time slice of the data. While interesting, this requires enough data in each time slice such that a high quality embedding can be trained for each. Further, because each time slice is trained independently, the dimensions of the embeddings are not comparable across time; they must use initialization (Kim et al., 2014) or ad-hoc alignment techniques (Kulkarni et al., 2015; Hamilton et al., 2016; Zhang et al., 2016) to stitch them together.

In contrast, the representations of dynamic embeddings are sequential latent variables. Dynamic embeddings naturally accommodates time slices with sparse data and immediately connect the latent dimensions across time. In Section 3, we found that dynamic embeddings provide quantitative improvements over independently fitting each slice.[1]

[1]Two similar models have been independently developed. Bamler & Mandt (2016) model both the embeddings and the context vectors using an Uhlenbeck-Ornstein process (Uhlenbeck & Ornstein, 1930). Yao et al. (2017) factorize the pointwise mutual information (PMI) matrix at different time slices. Their regularization also resembles an Uhlenbeck-Ornstein process.

Dynamic topic modeling also studies text data over time (Blei & Lafferty, 2006; Wang & McCallum, 2006; Wang et al., 2008; Gerrish & Blei, 2010; Wijaya & Yeniterzi, 2011; Yogatama et al., 2014; Mitra et al., 2014; 2015; Frermann & Lapata, 2016). This class of models describes documents in terms of topics, which are distributions over the vocabulary, and then allows the topics to change over the course of the collection. As in dynamic embeddings, some dynamic topic models use a Gaussian random walk to capture drift in the underlying language model; for example, see Blei & Lafferty (2006); Wang et al. (2008); Gerrish & Blei (2010); Frermann & Lapata (2016).

Though topic models and word embeddings are related, they are ultimately different approaches to language modeling. Topic models capture co-occurrence of words at the document level and focus on heterogeneity, i.e., that a document can exhibit multiple topics (Blei et al., 2003). Word embeddings capture co-occurrence in terms of proximity in the text, usually focusing on small neighborhoods around each word (Mikolov et al., 2013c). Combining dynamic topic models and dynamic word embeddings is an area for future study.

2. Dynamic Embeddings

We develop dynamic embeddings, a type of exponential family embedding (EFE) (Rudolph et al., 2016) that captures sequential changes in the representation of the data. We focus on text data and the Bernoulli embedding model.

In this section, we review Bernoulli embeddings for text and show how to include dynamics into the model. We then derive the objective function for dynamic embeddings and develop stochastic gradients to optimize it.

Bernoulli embeddings for text. An exponential family embedding is a conditionally specified model. It has three ingredients: The *context*, the *conditional distribution* of each data point, and the *parameter sharing structure*.

In an EFE for text, the data is a corpus of text. It is a collection of words $(x_1, \ldots, x_N)$ from a vocabulary of size V. Each word $x_i \in \{0,1\}^V$ is an indicator vector (also called a "one-hot" vector). It has exactly one nonzero entry at v, where v is the vocabulary term at position i.

In an EFE each data point has a *context*. In text, the context of each word is its neighborhood; thus it is modelled conditional on the words that come before and after it. Typical context sizes range between 2 and 10 words. (This is set in advance or by cross-validation.)

We will build on Bernoulli embeddings, which provide a conditional model for the individual entries of the indicator vectors $x_{iv} \in \{0,1\}$. Let c_i be the set of positions in the

neighborhood of position i and let $\mathbf{x}_{c_i}$ denote the collection of data points indexed by those positions. The conditional distribution of x_{iv} is

$$x_{iv}|\mathbf{x}_{c_i} \sim \text{Bern}(p_{iv}), \tag{1}$$

where $p_{iv} \in (0, 1)$ is the Bernoulli probability.[2]

Bernoulli embeddings specify the natural parameter, which is the log odds $\eta_{iv} = \log \frac{p_{iv}}{1-p_{iv}}$. It is a function of the representation of term v and the terms in the context of position i. Specifically, each index (i, v) in the data is associated with two parameter vectors, the *embedding vector* $\rho_v \in \mathbb{R}^K$ and the *context vector* $\alpha_v \in \mathbb{R}^K$. Together, the embedding vectors and context vectors form the natural parameter of the Bernoulli. It is

$$\eta_{iv} = \rho_v^\top \left(\sum_{j \in c_i} \sum_{v'} \alpha_{v'} x_{jv'} \right). \tag{2}$$

This is the inner product between the embedding ρ_v and the context vectors of the words that surround position i. (Because x_j is an indicator vector, the sum over the vocabulary selects the appropriate context vector α at position j.) The goal is to learn the embeddings and context vectors.

The index on the parameters does not depend on position i, but only on term v; the embeddings are shared across all positions in the text. This is what Rudolph et al. (2016) call the *parameter sharing structure*. It ensures, for example, that the embedding vector for INTELLIGENCE is the same wherever it appears in the corpus. (Dynamic embeddings partially relax this restriction.)

Finally, Rudolph et al. (2016) regularize the Bernoulli embedding by placing priors on the embedding and context vectors. They use Gaussian priors with diagonal covariance, i.e., ℓ_2 regularization. Without the regularization, fitting a Bernoulli embedding closely relates to other embedding techniques such as CBOW (Mikolov et al., 2013a) and negative sampling (Mikolov et al., 2013b). But the probabilistic perspective of Rudolph et al. (2016)—and in particular the priors and the parameter sharing—allows us to extend this setting to capture dynamics.

Dynamic Bernoulli embeddings. Dynamic Bernoulli embeddings extend Bernoulli embeddings to text data over time. Each observation x_{iv} is associated with a time slice t_i, such as the year of the observation. Context vectors are shared across all positions in the text but the embedding vectors are only shared within a time slice. Thus dynamic embeddings posit a sequence of embeddings for each term $\rho_v^{(t)} \in \mathbb{R}^K$.

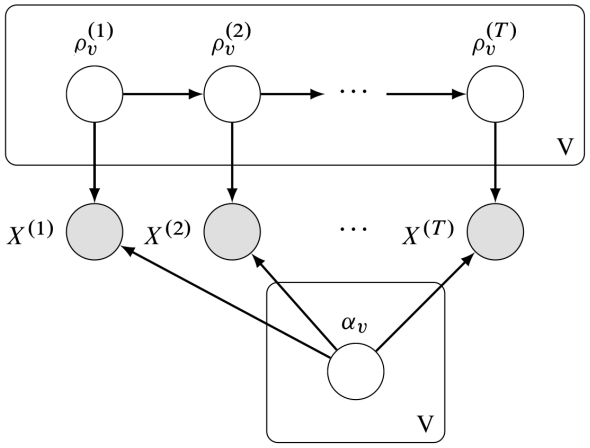


Figure 2. Graphical representation of a dynamic embedding for text data in T time slices, $X^{(1)}, \cdots, X^{(T)}$. The embedding vectors ρ_v of each term evolve over time. The context vectors are shared across all time slices.

The natural parameter of the conditional likelihood is similar to Equation (2) but with the embedding vector ρ_v replaced by the per-time-slice embedding vector $\rho_v^{(t_i)}$,

$$\eta_{iv} = \rho_v^{(t_i)\top} \left(\sum_{j \in c_j} \sum_{v'} \alpha_{v'} x_{jv'} \right). \tag{3}$$

Finally, dynamic embeddings use a Gaussian random walk as a prior on the embedding vectors,

$$\alpha_v, \rho_v^{(0)} \sim \mathcal{N}(0, \lambda_0^{-1} I) \tag{4}$$

$$\rho_v^{(t)} \sim \mathcal{N}(\rho_v^{(t-1)}, \lambda^{-1} I). \tag{5}$$

Given data, this leads to smoothly changing estimates of each term's embedding.[3]

Figure 2 gives the graphical model for dynamic embeddings. Dynamic embeddings are a conditionally specified model, which in general are not guaranteed to imply a consistent joint distribution. But dynamic Bernoulli embeddings model binary data, and thus a joint exists (Arnold et al., 2001).

Fitting dynamic embeddings. Calculating the joint is computationally intractable. Rather, we fit dynamic embeddings with the *pseudo log likelihood*, the sum of the log conditionals. This is a commonly used objective for conditionally specified models (Arnold et al., 2001).

In detail, we regularize the pseudo log likelihood with the log

[2] Multinomial embeddings (Rudolph et al., 2016) model each indicator vector x_i with a categorical conditional distribution, but this requires expensive normalization in form of a softmax function. For computational efficiency, one can replace the softmax with the hierarchical softmax (Mikolov et al., 2013b) or employ approaches related to noise contrastive estimation (Gutmann & Hyvärinen, 2010; Mnih & Kavukcuoglu, 2013). Bernoulli embeddings relax the one-hot constraint of x_i, and work well in practice; they relate to the negative sampling (Mikolov et al., 2013b).

[3] Because $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$ appear only as inner products in Equation (2), there is some redundancy in placing temporal dynamics on both the embeddings and the context vectors. Exploring dynamics in $\boldsymbol{\alpha}$ is a subject for future study.

priors and then maximize to obtain a pseudo MAP estimate. For dynamic Bernoulli embeddings, this objective is the sum of the log priors and the conditional log likelihoods of the data x_{iv}. We divide the data likelihood into two parts, the contribution of nonzero data entries $\mathcal{L}_{\text{pos}}$ and contribution of zero data entries $\mathcal{L}_{\text{neg}}$,

$$\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\alpha}) = \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{neg}} + \mathcal{L}_{\text{prior}}. \tag{6}$$

The likelihoods are

$$\mathcal{L}_{\text{pos}} = \sum_{i=1}^{N}\sum_{v=1}^{V} x_{iv} \log \sigma(\eta_{iv})$$
$$\mathcal{L}_{\text{neg}} = \sum_{i=1}^{N}\sum_{v=1}^{V} (1 - x_{iv}) \log(1 - \sigma(\eta_{iv})),$$

where $\sigma(\cdot)$ is the sigmoid, which maps natural parameters to probabilities.

The prior is

$$\mathcal{L}_{\text{prior}} = \log p(\boldsymbol{\alpha}) + \log p(\boldsymbol{\rho}),$$

where

$$\log p(\boldsymbol{\alpha}) = -\frac{\lambda_0}{2}\sum_{v} ||\alpha_v||^2$$
$$\log p(\boldsymbol{\rho}) = -\frac{\lambda_0}{2}\sum_{v} ||\rho_v^{(0)}||^2 - \frac{\lambda}{2}\sum_{v,t} ||\rho_v^{(t)} - \rho_v^{(t-1)}||^2.$$

The parameters $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}$ appear in the natural parameters η_{iv} of Equations (2) and (3) and in the log prior. The random walk prior penalizes consecutive word vectors $\rho_v^{(t-1)}$ and $\rho_v^{(t)}$ for drifting too far apart. It prioritizes parameter settings for which the norm of their difference is small.

The most expensive term in the objective is $\mathcal{L}_{\text{neg}}$, the contribution of the zeroes to the conditional log likelihood. The objective is cheaper if we subsample the zeros. Rather than summing over all words which are not at position i, we sum over a subset of negative examples drawn at random. Mikolov et al. (2013b) call this negative sampling and recommend sampling from the unigram distribution raised to the power of 0.75.

With negative sampling, we redefine $\mathcal{L}_{\text{neg}}$ in Equation (6). Denote the sampling distribution of zeros as $\hat{p}$,

$$\mathcal{L}_{\text{neg}} = \sum_{i=1}^{N}\sum_{v\sim\hat{p}} \log(1 - \sigma(\eta_{iv})). \tag{7}$$

This sum has fewer terms and reduces the contribution of the zeros to the objective. In a sense, this incurs a bias—the expectation with respect to the negative samples is not equal to the original objective—but "downweighting the zeros" can improve prediction accuracy (Hu et al., 2008; Liang et al., 2016).

Table 1. Time range and size of the three corpora analyzed in Section 3.

	time range	slices	slice size	vocab size	words
Arxiv ML	$2007 - 2015$	9	1 year	50k	6.5M
ACM	$1951 - 2014$	64	1 year	25k	21.6M
Senate speeches	$1858 - 2009$	76	2 years	25k	13.7M

We fit the objective (Equation (6) with Equation (7)) using stochastic gradients (Robbins & Monro, 1951) and with adaptive learning rates (Duchi et al., 2011). Pseudo code is in Appendix B. To avoid deriving the gradients of Equation (6), we implemented the algorithm in Edward (Tran et al., 2016). Edward is based on tensorflow (Team, 2015) and employs automatic differentiation.[4]

3. Empirical Study

Our empirical study contains two parts. In a quantitative evaluation we benchmark dynamic embeddings against static embeddings (Mikolov et al., 2013a;b; Rudolph et al., 2016). Dynamic embeddings improve over static embeddings in terms of the conditional likelihood of held-out predictions. Further, dynamic embeddings perform better than embeddings trained on the individual time slices (Hamilton et al., 2016). In a qualitative evaluation we use a fitted dynamic embedding model to extract which word vectors change most and we visualize their dynamics. Dynamic embeddings provide a new window into how language changes.

3.1. Data

We studied three datasets. See Table 1.

Machine Learning Papers (2007 - 2015): This dataset contains the full text from all machine learning papers (tagged "stat.ML") published on the Arxiv between April 2007 and June 2015. It spans 9 years and we treat each year as a time slice. The number of Arxiv papers about machine learning has increased over the years. There were 101 papers in 2007; there were $1,573$ papers in 2014.

Computer Science Abstracts (1951 - 2014): This dataset contains abstracts of computer science papers published by the Association of Computing Machinery (ACM) from 1951 to 2014. Again, each year is considered a time slice and here too the amount of data increases over the years. For 1953, there are only around 10 abstracts and their combined length

[4]Code available at `http://github.com/mariru/dynamic_bernoulli_embeddings`

is only 471 words; the combined length of the abstracts from 2009 is over 2M.

Senate Speeches (1858 - 2009): This dataset contains all U.S. Senate speeches from 1858 to mid 2009. Here we treat every 2 years as a time slice. In contrast to the other datasets, this corpus is a transcript of spoken language.

For all datasets, we divide the observations into training, validation, and testing. Within each time slice we use 80% for training, 10% for validation, and 10% for testing. Appendix A provides details about preprocessing.

3.2. Quantitative evaluation

We compare dynamic embeddings (D-EMB) to time-binned embeddings (T-EMB) (Hamilton et al., 2016) and static embeddings (S-EMB) (Rudolph et al., 2016). There are many embedding techniques, without dynamics, that enjoy comparable performance. For the S-EMB, we study Bernoulli embeddings (Rudolph et al., 2016), which are similar to continuous bag-of-words (CBOW) with negative sampling (Mikolov et al., 2013a;b). For time-binned embeddings, Hamilton et al. (2016) train a separate embedding on each time slice.

Evaluation metric. We evaluate models by held-out Bernoulli probability. Given a model, each held-out word (validation or testing) is associated with a Bernoulli probability. At that position, a better model assigns higher probability to the observed word and lower probability to the others. This metric is straightforward because the competing methods all produce Bernoulli conditional likelihoods (Equation (1)).[5] We report $\mathcal{L}_{\text{pos}}$, which considers only the nonzero held-out data. To make results comparable, all methods are trained with the same number of negative samples.

Model training and hyperparameters. Each method takes a maximum of 10 passes over the data. (The corresponding number of stochastic gradient steps depends on the size of the minibatches.) The parameters of S-EMB are initialized randomly. We initialize both D-EMB and T-EMB from a fit of S-EMB which has been trained from one pass, and then train for 9 additional passes.

We set the dimension of the embeddings to 100 and the number of negative samples to 20. We experiment with two context sizes, 2 and 8.

Other parameters are set by validation error. All methods use validation error to set the initial learning rate η and minibatch sizes m. The model selects $\eta \in [0.01, 0.1, 1, 10]$ and $m \in [0.001N, 0.0001N, 0.00001N]$, where N is the size of training data. The only parameter specific to D-EMB is the precision of the random drift. To have one less hyper parameter to tune, we fix the precision on the context vectors and the initial dynamic embeddings to $\lambda_0 = \lambda/1000$, a constant multiple of the precision on the dynamic embeddings. We choose $\lambda \in [1, 10]$ by validation error.

Results. We train each model on each training set and use each validation set for model selection (e.g., selecting the minibatch size and learning rate). Table 2 reports the results on the test set. Dynamic embeddings consistently achieve higher held-out likelihood.

Table 2. Dynamic embeddings (D-EMB) consistently achieve highest held-out $\mathcal{L}_{\text{pos}}$ (Equation (6)). We compare to static embeddings (S-EMB) (Mikolov et al., 2013b; Rudolph et al., 2016), time-binned embeddings (T-EMB) (Hamilton et al., 2016).

Arxiv ML

	context size 2	context size 8
S-EMB (Rudolph et al., 2016)	-2.706 ± 0.002	-2.491 ± 0.002
T-EMB (Hamilton et al., 2016)	-2.646 ± 0.002	-2.454 ± 0.002
D-EMB [this paper]	$\mathbf{-2.535 \pm 0.001}$	$\mathbf{-2.400 \pm 0.002}$

Senate speeches

	context size 2	context size 8
S-EMB (Rudolph et al., 2016)	-2.366 ± 0.001	-2.244 ± 0.001
T-EMB (Hamilton et al., 2016)	-2.295 ± 0.001	-2.212 ± 0.001
D-EMB [this paper]	$\mathbf{-2.263 \pm 0.001}$	$\mathbf{-2.204 \pm 0.001}$

ACM

	context size 2	context size 8
S-EMB (Rudolph et al., 2016)	-2.427 ± 0.001	-2.231 ± 0.001
T-EMB (Hamilton et al., 2016)	-2.420 ± 0.001	-2.242 ± 0.001
D-EMB [this paper]	$\mathbf{-2.396 \pm 0.001}$	$\mathbf{-2.228 \pm 0.001}$

3.3. Qualitative exploration

We now show how to use dynamic embeddings to explore the dataset. We use the fitted model to suggest ways that language changes and visualize its discovered dynamic structure.

A word's *embedding neighborhood* helps visualize its usage and how it changes over time. It is simply a list of other words with similar usage. For a given query word (e.g., COMPUTER) we take its index v and select the top ten words according to

$$\text{neighborhood}(v, t) = \text{argsort}_w \Big(\frac{\text{sign}(\rho_v^{(t)})^\top \rho_w^{(t)}}{||\rho_v^{(t)}|| \cdot ||\rho_w^{(t)}||} \Big). \quad (8)$$

We fit a dynamic embedding fit to the Senate speeches. Table 3 gives the embedding neighborhoods of COMPUTER for the years 1858 and 1986. Its usage changed dramatically over the years. In 1858, a COMPUTER was a profession, a person who was hired to compute things. Now the profession is obsolete; COMPUTER refers to the electronic device.

Table 3 provides another example, BUSH. In 1858 this word always referred to the plant. A BUSH still is a plant, but

[5] Since we hold out chunks of consecutive words usually both a word and its context are held out. For all methods we have to use the words in the context to compute the conditional likelihoods.

Table 3. Embedding neighborhoods (Equation (8)) reveal how the usage of a word changes over time. The embedding neighborhoods of COMPUTER and BUSH were computed from a dynamic embedding fitted to Congress speeches (1858-2009). COMPUTER used to be a profession but today it is used to refer to the electonic device. The word BUSH is a plant but eventually in congress BUSH is used to refer to the political figures. The embedding neighborhood of DATA comes from a dynamic embedding fitted to ACM abstracts (1951-2014).

COMPUTER (Senate)

1858	1986
computer	computer
draftsman	software
draftsmen	computers
copyist	copyright
photographer	technological
computers	innovation
copyists	mechanical
janitor	hardware
accountant	technologies
bookkeeper	vehicles

BUSH (Senate)

1858	1990
bush	bush
barberry	cheney
rust	nonsense
bushes	nixon
borer	reagan
eradication	george
grasshoppers	headed
cancer	criticized
tick	clinton
eradicate	blindness

DATA (ACM)

1961	1969	1991	2011	2014
data	data	data	data	data
directories	repositories	voluminous	raw data	data streams
files	voluminous	raw data	voluminous	voluminous
bibliographic	lineage	repositories	data sources	raw data
formatted	metadata	data streams	data streams	warehouses
retrieval	snapshots	data sources	dws	dws
publishing	data streams	volumes	repositories	repositories
archival	raw data	dws	warehouses	data sources
archives	cleansing	dsms	marts	data mining
manuscripts	data mining	data access	volumes	marts

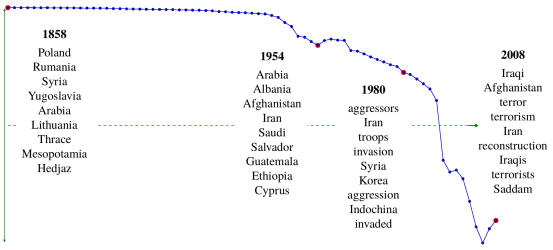


Figure 3. The dynamic embedding captures how the usage of the word IRAQ changes over the years (1858-2009). The x-axis is time and the y-axis is a one-dimensional projection of the embeddings using PCA. We include the embedding neighborhoods for IRAQ in the years 1858, 1954, 1980 and 2008.

Table 4. A list of the top 16 words whose dynamic embedding on Senate speeches changes most. The number represents the absolute drift (Equation (9)). The dynamics of the capitalized words are in Table 5 and discussed in the main text.

words with largest drift (Senate)

IRAQ	3.09	coin	2.39
tax cuts	2.84	social security	2.38
health care	2.62	FINE	2.38
energy	2.55	signal	2.38
medicare	2.55	program	2.36
DISCIPLINE	2.44	moves	2.35
text	2.41	credit	2.34
VALUES	2.40	UNEMPLOYMENT	2.34

in the 1990's, in the Senate, it is usually refers to political figures. Unlike COMPUTER, where the embedding neighborhoods reveal two mutually exclusive meanings, the embedding neighborhoods of BUSH reflect which meaning is more prevalent in a given period.

A final example in Table 3 is the word DATA, from the ACM abstracts. The evolution of the embedding neighborhoods of DATA reflects how its meaning changes in the computer science literature.

Finding changing words with absolute drift. We have highlighted example words whose usage changes. However, not all words have changing usage. We now define a metric to discover which words change most.

One way to find words that change is to use *absolute drift*. For word v, it is

$$\text{drift}(v) = ||\rho_v^{(T)} - \rho_v^{(0)}||. \tag{9}$$

This is the Euclidean distance between the word's embedding at the last time slice and at the first time slice.

In the Senate speeches, Table 4 shows the 16 words that have largest absolute drift. The word IRAQ has largest drift. Figure 3 highlights IRAQ's embedding neighborhood in four time slices: 1858, 1950, 1980, and 2008. (Appendix C gives the entire trajectory of its embedding neighborhood.) At first the neighborhood contains other countries and regions. Later, Arab countries move to the top of the neighborhood, suggesting that the speeches start to use rhetoric more specific to Arab countries. In 1980, Iraq invades Iran and the Iran-Iraq war begins. In these years words such as AGGRESSORS, TROOPS, and INVASION appear in the embedding neighborhood. Eventually, by 2008, the neighborhood contains TERROR, TERRORISM, and SADDAM.

Four other words with large drift are DISCIPLINE, VALUES, FINE and UNEMPLOYMENT (Table 4). Table 5 shows their embedding neighborhoods for selected years. Of these words, DISCIPLINE, VALUES and, FINE have multiple meanings. Their neighborhoods reflect how the dominant meaning changes over time. For example, VALUES can be either a numerical quantity or can be used to refer to moral values and principles. In contrast, IRAQ and UNEMPLOYMENT are both words which have always had the same definition. Yet, the evolution of their neighborhood captures changes in the way they are used.

Dynamic embeddings as a tool to study a text. Our hope is that dynamic embeddings provide a suggestive tool for understanding change in language. For example, researchers interested in UNEMPLOYMENT can complement their investigation by looking at the embedding neighborhood of related

Table 5. Embedding neighborhoods extracted from a dynamic embedding fitted to Senate speeches (1858 - 2009). DISCIPLINE, VALUES, FINE, and UNEMPLOYMENT are within the 16 words whose dynamic embedding has largest absolute drift. (Table 4).

DISCIPLINE		VALUES		FINE		UNEMPLOYMENT		
1858	**2004**	**1858**	**2000**	**1858**	**2004**	**1858**	**1940**	**2000**
discipline	discipline	values	values	fine	fine	unemployment	unemployment	unemployment
hazing	balanced	fluctuations	sacred	luxurious	punished	unemployed	unemployed	jobless
westpoint	balancing	value	inalienable	finest	penitentiaries	depression	depression	rate
assaulting	fiscal	currencies	unique	coarse	imprisonment	acute	alleviating	depression
disciplined	let	fluctuation	preserving	beautiful	misdemeanor	deplorable	destitution	forecasts
courtmartial	ourselves	depreciation	exemplified	imprisonment	punishable	alleviating	acute	crate
punishment	structural	fluctuating	principles	finer	offense	destitution	reemployment	upward
martial	deficit	purchasing power	philanthropy	lighter	guilty	urban	deplorable	lag
mentally	administrations	fluctuate	virtues	weaves	conviction	employment	employment	economists
summarily	restraint	basis	historical	spun	penitentiary	distressing	distress	predict

Table 6. Using dynamic embeddings we can study a social phenomenon of interest. We pick a target word of interest, such as JOBS or PROSTITUTION and create their embedding neighborhoods (Equation (8)). Looking at the neighborhood of JOBS complements the evolution of UNEMPLOYMENT (Table 5). Or we might want to study PROSTITUTION. It used to be considered immoral and vile, evolved to be indecent and its neighborhood in 1990 reveals a more concerned outlook as it includes words like SERVITUDE, HARASSMENT and TRAFFICKING.

JOBS			PROSTITUTION					
1858	**1938**	**2008**	**1858**	**1930**	**1945**	**1962**	**1988**	**1990**
jobs	jobs	jobs	prostitution	prostitution	prostitution	prostitution	harassment	prostitution
employment	unemployed	job	punishing	punishing	indecent	indecent	intimidation	servitude
unemployed	employment	create	immoral	immoral	vile	harassment	prostitution	harassment
overtime	job	creating	illegitimate	bootlegging	immoral	intimidation	counterfeit	intimidation
positions	overtime	tremendously	riotous	riotous	induces	sexual	illegal	trafficking
job	positions	economies	mobs	forbidden	incite	vile	trafficking	harassing
idleness	shifts	opportunities	violence	anarchists	abortion	counterfeit	indecent	apprehended
working	idleness	created	assemblage	assemblage	forbid	anarchists	disregard	killings
busy	busy	pace	criminals	forbid	harboring	mobs	anarchists	labeled
civil service	salaried	michigan	procures	abet	assemblage	lawbreakers	punishing	naked

words such as EMPLOYMENT, JOBS or LABOR. In Table 6 we list the neighborhoods of JOBS for the years 1858, 1938, and 2008. In 2008 the embedding neighborhood contains words like CREATE and OPPORTUNITIES, suggesting a different outlook on JOBS than in earlier years.

Another interesting example is PROSTITUTION. It used to be IMMORAL and VILE, went to INDECENT, and in modern days it is considered HARASSMENT. We note the word PROSTITUTION is not a frequent word. On average, it is used once per time slice and, in two thirds of the time slices, it is not mentioned at all. Yet, the model is able to learn about PROSTITUTION and the temporal evolution of the embedding neighborhood reveals how over the years a judgemental stance turns into concern over a social issue.

4. Summary

We described dynamic embeddings, distributed representations of words that drift over the course of the collection. Building on Rudolph et al. (2016), we formulate word embeddings with conditional probabilistic models and then incorporate dynamics with a Gaussian random walk prior. We fit dynamic embeddings with stochastic optimization.

We used dynamic embeddings to analyze several datasets: 8 years of machine learning papers, 63 years of computer science abstracts, and 151 years of speeches in the U.S. Senate. Dynamic embeddings provided a better fit than static embeddings and other methods that account for time.

Finally, we demonstrated how dynamic embeddings can help identify interesting ways that language changes. A word's meaning can change (e.g., COMPUTER); its dominant meaning can change (e.g., VALUES); or its related subject matter can change (e.g., IRAQ).

Acknowledgements

We would like to thank Francisco Ruiz and Liping Liu for discussion and helpful suggestiongs, Elliot Ash and Suresh Naidu for access to the Congress speeches, and Aaron Plasek and Matthew Jones for access to the ACM abstracts.

References

Aitchison, Jean. *Language change: progress or decay?* Cambridge University Press, 2001.

Arnold, Barry C, Castillo, Enrique, Sarabia, Jose Maria, et al. Conditionally specified distributions: an introduction (with comments and a rejoinder by the authors). *Sta-*

tistical Science, 16(3):249–274, 2001.

Arora, Sanjeev, Li, Yuanzhi, Liang, Yingyu, Ma, Tengyu, and Risteski, Andrej. Rand-walk: A latent variable model approach to word embeddings. *arXiv preprint arXiv:1502.03520*, 2015.

Bamler, Robert and Mandt, Stephan. Dynamic word embeddings via skip-gram filtering. *arXiv preprint arXiv:1702.08359*, 2016.

Basile, Pierpaolo, Caputo, Annalina, and Semeraro, Giovanni. Analysing word meaning over time by exploiting temporal random indexing. In *First Italian Conference on Computational Linguistics CLiC-it*, 2014.

Bengio, Yoshua, Ducharme, Réjean, Vincent, Pascal, and Jauvin, Christian. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Bishop, Christopher M. Machine learning and pattern recognition. *Information Science and Statistics. Springer, Heidelberg*, 2006.

Blei, David M and Lafferty, John D. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pp. 113–120. ACM, 2006.

Blei, David M, Ng, Andrew Y, and Jordan, Michael I. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (Jul):2121–2159, 2011.

Frermann, Lea and Lapata, Mirella. A bayesian model of diachronic meaning change. *Transactions of the Association for Computational Linguistics*, 4:31–45, 2016.

Gerrish, S. and Blei, D. A language-based approach to measuring scholarly impact. In *International Conference on Machine Learning*, 2010.

Gutmann, Michael and Hyvärinen, Aapo. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

Hamilton, William L, Leskovec, Jure, and Jurafsky, Dan. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*, 2016.

Harris, Zellig S. Distributional structure. *Word*, 10(2-3): 146–162, 1954.

Hu, Yifan, Koren, Yehuda, and Volinsky, Chris. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 263–272. Ieee, 2008.

Kim, Yoon, Chiu, Yi-I, Hanaki, Kentaro, Hegde, Darshan, and Petrov, Slav. Temporal analysis of language through neural language models. *arXiv preprint arXiv:1405.3515*, 2014.

Kirby, Simon, Dowman, Mike, and Griffiths, Thomas L. Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences*, 104(12): 5241–5245, 2007.

Kulkarni, Vivek, Al-Rfou, Rami, Perozzi, Bryan, and Skiena, Steven. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 625–635. ACM, 2015.

Levy, Omer and Goldberg, Yoav. Neural word embedding as implicit matrix factorization. In *Neural Information Processing Systems*, pp. 2177–2185, 2014.

Liang, Dawen, Charlin, Laurent, McInerney, James, and Blei, David M. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 951–961. International World Wide Web Conferences Steering Committee, 2016.

Mihalcea, Rada and Nastase, Vivi. Word epoch disambiguation: Finding how words change over time. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pp. 259–263. Association for Computational Linguistics, 2012.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. *ICLR Workshop Proceedings. arXiv:1301.3781*, 2013a.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*, pp. 3111–3119, 2013b.

Mikolov, Tomas, Yih, Wen-T au, and Zweig, Geoffrey. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pp. 746–751, 2013c.

Mitra, Sunny, Mitra, Ritwik, Riedl, Martin, Biemann, Chris, Mukherjee, Animesh, and Goyal, Pawan. That's sick dude!: Automatic identification of word sense change across different timescales. *arXiv preprint arXiv:1405.4392*, 2014.

Mitra, Sunny, Mitra, Ritwik, Maity, Suman Kalyan, Riedl, Martin, Biemann, Chris, Goyal, Pawan, and Mukherjee,

Animesh. An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering*, 21(05):773–798, 2015.

Mnih, Andriy and Kavukcuoglu, Koray. Learning word embeddings efficiently with noise-contrastive estimation. In *Neural Information Processing Systems*, pp. 2265–2273, 2013.

Murphy, Kevin P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. In *Conference on Empirical Methods on Natural Language Processing*, volume 14, pp. 1532–1543, 2014.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Rudolph, Maja, Ruiz, Francisco, Mandt, Stephan, and Blei, David. Exponential family embeddings. In *Advances in Neural Information Processing Systems*, pp. 478–486, 2016.

Rumelhart, David E, Hintont, Geoffrey E, and Williams, Ronald J. Learning representations by back-propagating errors. *Nature*, 323:9, 1986.

Sagi, Eyal, Kaufmann, Stefan, and Clark, Brady. Tracing semantic change with latent semantic analysis. *Current methods in historical semantics*, pp. 161–183, 2011.

Tang, Xuri, Qu, Weiguang, and Chen, Xiaohe. Semantic change computation: A successive approach. *World Wide Web*, 19(3):375–415, 2016.

Team, Tensorflow. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

Tran, Dustin, Kucukelbir, Alp, Dieng, Adji B., Rudolph, Maja, Liang, Dawen, and Blei, David M. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

Uhlenbeck, George E and Ornstein, Leonard S. On the theory of the brownian motion. *Physical review*, 36(5): 823, 1930.

Wang, C., Blei, D., and Heckerman, D. Continuous time dynamic topic models. In *Uncertainty in Artificial Intelligence (UAI)*, 2008.

Wang, Xuerui and McCallum, Andrew. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 424–433. ACM, 2006.

Wijaya, Derry Tanti and Yeniterzi, Reyyan. Understanding semantic change of words over centuries. In *Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web*, pp. 35–40. ACM, 2011.

Yao, Zijun, Sun, Yifan, Ding, Weicong, Rao, Nikhil, and Xiong, Hui. Discovery of evolving semantics through dynamic word embedding learning. *arXiv preprint arXiv:1703.00607*, 2017.

Yogatama, D., Wang, C., Routledge, B., Smith, N. A, and Xing, E. Dynamic language models for streaming text. *Transactions of the Association for Computational Linguistics*, 2:181–192, 2014.

Zhang, Yating, Jatowt, Adam, Bhowmick, Sourav S, and Tanaka, Katsumi. The past is not a foreign country: Detecting semantically similar terms across time. *IEEE Transactions on Knowledge and Data Engineering*, 28 (10):2793–2807, 2016.

A. Data Preprocessing

We fix the vocabulary to the 25000 most frequent words and remove all words from the documents which are not in the vocabulary. As in (Mikolov et al., 2013b) we additionally remove each word with probability $p = 1 - \sqrt{(\frac{10^{-5}}{f_i})}$ where f_i is the frequency of the word. This effectively downsamples especially the frequent words and speeds up training. From each time slice 80% of the words are used for training. A random subsample of 10% of the words is held out for validation and another 10% for testing.

B. Pseudo code

Algorithm 1: Minibatch stochastic gradient descent for dynamic Bernoulli embeddings.

Input: T time slices of text data $X^{(t)}$ of size m_t respectively. Context size c, size of embedding K, number of negative samples n, number of minibatch fractions m, initial learning rate η, precision λ, vocabulary size V, smoothed unigram distribution $\hat{p}$.

for $v = 1$ **to** V **do**
  Initialize entries of α_v
  (using draws from a normal distribution with zero mean and standard deviation 0.01).
  for $v = 1$ **to** V **do**
    Initialize entries of $\rho_v^{(t)}$
    (using draws from a normal distribution with zero mean and standard deviation 0.01).
  end for
end for
for number of passes over the data **do**
  for number of minibatch fractions m **do**
    for $t = 1$ **to** T **do**
      Sample minibatch of m_t/m consecutive words $\{x_1^{(t)}, \cdots, x_{m_t/m}^{(t)}\}$ from each time slice $X^{(t)}$, and use each word's context to construct

$$C_i^{(t)} = \sum_{j \in c_i} \sum_{v'=1}^{V} \alpha_{v'} x_{jv'}. \tag{10}$$

      For each text position in the minibatch, draw a set $\mathcal{S}_i^{(t)}$ of n negative samples from $\hat{p}$
    end for
    update the parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\rho}\}$ by ascending the stochastic gradient

$$\nabla_{\boldsymbol{\theta}} \Bigg\{ \sum_{t=1}^{T} m \sum_{i=1}^{m_t/m} \Big(\sum_{v=1}^{V} x_{iv}^{(t)} \log \sigma(\rho_v^{(t)\top} C_i^{(t)}) + \sum_{x_j \in \mathcal{S}_i^{(t)}} \sum_{v=1}^{V} (1 - x_{jv}) \log(1 - \sigma(\rho_v^{(t)\top} C_i^{(t)})) \Big)$$
$$- \frac{\lambda_0}{2} \sum_v ||\alpha_v||^2 - \frac{\lambda_0}{2} \sum_v ||\rho_v^{(0)}||^2 - \frac{\lambda}{2} \sum_{v,t} ||\rho_v^{(t)} - \rho_v^{(t-1)}||^2 \Bigg\}$$

  end for
end for

Any standard gradient-based learning rate schedule can be used. We use Adagrad (Duchi et al., 2011) in our experiments.

C. Entire Embedding Trajectory of IRAQ

Here we give the entire trajectory of the embedding neighborhood of IRAQ. Over the years it drifts smoothly. On average IRAQ is mentioned only 10.6 times per time slice and in 64 out of the 76 time slices, IRAQ is not even mentioned at all. For these years, the prior (Equation (4)) ensures that the embedding at time t is the average of the embeddings at time $t-1$ and $t+1$. When the embedding vector does not change between two consecutive time slices, the embedding neighborhood might still fluctuate. This is because computing the embedding neighborhoods (Equation (8)) involves also the embedding vectors of the other words in the vocabulary.

Table 7. Embedding neighborhood of IRAQ extracted from a dynamic embedding fitted to the congress data. It is the word whose embedding vector has largest absolute drift. By listing the neighborhood for all the time bins, we can see how Iraq's embedding vector drifts smoothly. In 1858 IRAQ's embedding neighborhood contains countries and regions. In 1950 a rethoric more specific to arabic countries crystalizes. In 1980 Iraq invades Iran and words like invasion, aggressor and troops are in the neighborhood. By 2008 the embedding neighborhood of iran contains words like terror, terrorism and saddam.

1858	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1860	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1862	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1864	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1866	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1868	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1870	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1872	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1874	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1876	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1878	poland, rumania, syria, yugoslavia, arabia, lithuania, thrace, mesopotamia, hedjaz, albania
1880	poland, rumania, syria, yugoslavia, arabia, thrace, lithuania, mesopotamia, hedjaz, albania
1882	poland, rumania, yugoslavia, syria, arabia, thrace, lithuania, mesopotamia, hedjaz, albania
1884	rumania, poland, yugoslavia, syria, arabia, thrace, lithuania, mesopotamia, hedjaz, albania
1886	rumania, poland, yugoslavia, syria, arabia, thrace, lithuania, mesopotamia, hedjaz, albania
1888	rumania, poland, yugoslavia, syria, arabia, thrace, lithuania, mesopotamia, hedjaz, albania
1890	rumania, poland, yugoslavia, arabia, syria, thrace, lithuania, mesopotamia, hedjaz, albania
1892	rumania, poland, yugoslavia, arabia, syria, thrace, lithuania, mesopotamia, hedjaz, albania
1894	rumania, yugoslavia, arabia, poland, syria, thrace, lithuania, mesopotamia, hedjaz, albania
1896	rumania, yugoslavia, arabia, syria, poland, thrace, lithuania, mesopotamia, hedjaz, albania
1898	rumania, yugoslavia, arabia, syria, poland, thrace, lithuania, mesopotamia, hedjaz, albania
1900	rumania, yugoslavia, arabia, syria, poland, thrace, mesopotamia, lithuania, albania, hedjaz
1902	rumania, yugoslavia, arabia, syria, poland, thrace, mesopotamia, lithuania, albania, hedjaz
1904	rumania, yugoslavia, arabia, syria, poland, thrace, mesopotamia, albania, lithuania, hedjaz
1906	rumania, yugoslavia, arabia, syria, poland, thrace, mesopotamia, albania, lithuania, hedjaz
1908	rumania, yugoslavia, arabia, syria, poland, thrace, mesopotamia, albania, hedjaz, lithuania
1910	syria, rumania, arabia, yugoslavia, thrace, mesopotamia, czecho, albania, poland, lithuania
1912	syria, rumania, arabia, yugoslavia, mesopotamia, thrace, czecho, albania, lithuania, poland
1914	syria, rumania, arabia, yugoslavia, thrace, mesopotamia, albania, czecho, poland, lithuania
1916	syria, rumania, arabia, yugoslavia, thrace, mesopotamia, albania, czecho, poland, lithuania
1918	rumania, syria, arabia, yugoslavia, poland, czecho, mesopotamia, lithuania, thrace, serbia
1920	rumania, syria, arabia, yugoslavia, poland, czecho, mesopotamia, thrace, lithuania, serbia
1922	rumania, syria, arabia, yugoslavia, poland, mesopotamia, czecho, thrace, lithuania, serbia
1924	rumania, syria, arabia, yugoslavia, poland, mesopotamia, czecho, thrace, lithuania, serbia
1926	rumania, arabia, syria, mesopotamia, yugoslavia, albania, thrace, salvador, persian, czecho
1928	arabia, rumania, syria, yugoslavia, mesopotamia, albania, thrace, persian, salvador, bulgaria
1930	arabia, rumania, syria, yugoslavia, mesopotamia, albania, thrace, persian, salvador, bulgaria
1932	arabia, rumania, syria, yugoslavia, mesopotamia, albania, thrace, salvador, persian, bulgaria
1934	arabia, rumania, syria, yugoslavia, albania, mesopotamia, thrace, salvador, persian, bulgaria
1936	arabia, rumania, syria, yugoslavia, albania, mesopotamia, thrace, salvador, persian, bulgaria
1938	rumania, arabia, syria, yugoslavia, albania, mesopotamia, bulgaria, salvador, alsace, lithuania
1940	rumania, arabia, syria, yugoslavia, mesopotamia, albania, salvador, guatemala, bulgaria, thrace
1942	arabia, yugoslavia, guatemala, albania, salvador, syria, rumania, bulgaria, iraq, balkans
1944	yugoslavia, salvador, iraq, guatemala, arabia, bulgaria, albania, rumania, syria, iran
1946	iraq, albania, arabia, salvador, guatemala, iran, bulgaria, afghanistan, rumania, syria
1948	iraq, albania, arabia, salvador, guatemala, iran, bulgaria, afghanistan, rumania, syria
1950	iraq, arabia, albania, afghanistan, iran, saudi, salvador, guatemala, ethiopia, cyprus
1952	iraq, arabia, saudi, albania, afghanistan, iran, salvador, guatemala, cyprus, ethiopia
1954	iraq, albania, bulgaria, arabia, iran, salvador, afghanistan, rumania, syria, cyprus
1956	iraq, albania, iran, bulgaria, syria, rumania, afghanistan, salvador, arabia, guatemala
1958	iran, syria, albania, afghanistan, iraq, bulgaria, arabia, rumania, cyprus, sultan
1960	iran, syria, albania, afghanistan, iraq, bulgaria, arabia, rumania, cyprus, sultan
1962	iran, iraq, syria, afghanistan, invasion, invaded, indochina, egypt, cyprus, arabia
1964	iran, syria, iraq, afghanistan, invasion, invaded, egypt, indochina, turkey, turkish
1966	iran, iraq, syria, invasion, afghanistan, invaded, indochina, korea, egypt, aggressors
1968	iraq, iran, syria, invasion, invaded, afghanistan, indochina, korea, aggressors, egypt
1970	iraq, iran, invasion, syria, invaded, afghanistan, korea, indochina, aggressors, egypt
1972	iraq, invasion, iran, syria, invaded, afghanistan, aggressors, korea, indochina, troops
1974	iraq, invasion, iran, korea, syria, invaded, troops, aggressors, afghanistan, indochina
1976	iraq, iran, aggressors, aggression, syria, invasion, troops, korea, invaded, indochina
1978	iraq, aggressors, troops, iran, invasion, syria, korea, aggression, indochina, invaded
1980	iraq, aggressors, iran, troops, invasion, syria, korea, aggression, indochina, invaded
1982	iraq, aggressors, iran, troops, invasion, syria, korea, aggression, indochina, invaded
1984	iraq, iran, aggressors, syria, invasion, troops, korea, aggression, invaded, allies
1986	iraq, aggressors, syria, invasion, iran, korea, troops, invaded, aggression, iraqi
1988	iraq, aggressors, invasion, iran, allies, invaded, korea, iraqi, syria, aggression
1990	iraq, iraqi, iran, afghanistan, invaded, aggressors, terror, allies, korea, invasion
1992	iraq, iran, terror, allies, iraqi, korea, aggressors, afghanistan, syria, invaded
1994	iraq, iraqi, invaded, korea, allies, aggressors, iran, exit, afghanistan, terror
1996	iraq, iran, iraqi, allies, afghanistan, terror, invaded, korea, aggressors, syria
1998	iraq, terror, iran, iraqi, afghanistan, occupation, allies, invaded, troops, invasion
2000	iraq, iraqi, afghanistan, terrorism, terror, iraqis, iran, reconstruction, saddam, bosnia
2002	iraq, iraqi, afghanistan, terrorism, iran, terror, iraqis, reconstruction, terrorist, terrorists
2004	iraq, iraqi, afghanistan, terror, terrorism, iran, reconstruction, iraqis, terrorists, saddam
2006	iraq, iraqi, afghanistan, terror, terrorism, iran, reconstruction, iraqis, terrorists, saddam
2008	iraq, iraqi, afghanistan, terror, terrorism, iran, reconstruction, iraqis, terrorists, saddam