MODELING EVOLUTION OF LANGUAGE THROUGH TIME WITH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language evolves over time with trends and shifts in technological, political, or cultural contexts. Capturing these variations is important to develop better language models. While recent works tackle temporal drifts by learning diachronic embeddings, we instead propose to integrate a temporal component into a recurrent language model. It takes the form of global latent variables, which are structured in time by a learned non-linear transition function. We perform experiments on three time annotated corpora. Experimental results on language modeling and classification tasks show that our model performs consistently better than temporal word embedding methods in two temporal evaluation settings: prediction and modeling. Moreover, we empirically show that the system is able to predict informative latent states in the future.

1 INTRODUCTION

Language modeling with deep neural networks is an active research field (Howard & Ruder, 2018; Melis et al., 2018; Merity et al., 2018a;b). It is a central task in Natural Language Processing (NLP) as it plays a major role in various text related tasks such as: speech recognition (Chiu et al., 2017), image captioning (Vinyals et al., 2017), or text generation (Fedus et al., 2018). The task is to predict the probability distribution of the next word in a text sequence. The standard deep architecture for language models (LMs) has been the Recurrent Neural Network (RNN) for several years (Mikolov et al., 2010). Moreover, their sophisticated version, LSTM networks (Hochreiter & Schmidhuber, 1997), are still state of the art in language modeling (Melis et al., 2018; Merity et al., 2018b), although research on new architectures is very active (Vaswani et al., 2017; Bai et al., 2018).

Currently, recurrent language models are static and do not consider the various shifts that affect language; meaning of words can shift, new words appear as other vanish, and yesterday's topics are different from tomorrow's. To handle temporal variations in texts, recent research mainly focus on learning distinct word embeddings per timestep (Kim et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016), and smoothing them in time (Bamler & Mandt, 2017; Yao et al., 2018). Indeed, word embeddings are powerful tools to capture and analyze semantic relations between word pairs (Mikolov et al., 2013). However, learning different embeddings for each timestep leads to learning algorithms with high time and memory complexity. This leads several approximations. For instance, Yao et al. (2018) use alternate optimization that breaks the flow of gradient through time. And Bamler & Mandt (2017) in their smoothing skip-gram algorithm propose complex gradient estimations that involves solving tridiagonal linear systems which cannot be parallelized in time.

In this paper, we propose a dynamic recurrent language model that uses global temporal variables instead of learning distinct embeddings at each timestep. Our contribution is threefold:

- We propose a dynamic recurrent language model in the form of an LSTM conditioned on global latent variables structured in time. By also learning a transition function in the latent space, we are able to predict states at future unseen timesteps. The learning procedure is straightforward thanks to the pathwise derivative (Kingma & Welling, 2014; Rezende et al., 2014), thus scales to large corpora easily.
- We adapt temporal word embeddings algorithms to recurrent language modeling. We empirically show that these methods are not fit for learning with RNNs, and that the temporal

embeddings almost systematically fail to beat non-temporal baselines on downstream classification tasks.

- We perform experiments on three time annotated text corpora with varying sizes and temporal scales. We evaluate the models on language modeling and on downstream classifications tasks.

2 RELATED WORK

Studying of language evolution has been of interest for a long time in machine learning and information retrieval communities. Topics detection and tracking were among the firsts approaches to study language evolution. In 2002, Kabán & Girolami (2002) proposed a Hidden Markov Model (HMM) to visualize temporal evolution of topics in a textual stream. In 2006, Wang & McCallum (2006) and Blei & Lafferty (2006) proposed non-Markovian models based on Latent Dirichlet Allocations (LDA). While Wang & McCallum (2006) learn distributions of topics through time, Blei & Lafferty (2006) learn word distributions conditioned on latent topics that evolve through time with a State Space Model. However, these methods require to manually tune the number of latent topics, and language models are limited to simple word occurrence distributions. Moreover, these models are usually limited to specific conjugate distributions on the latent variables to allow tractable Variational Inference. Note that Blei & Lafferty (2006) led to several extensions, for instance with multi-scale temporal variable (Iwata et al. (2012)), or continuous time dependencies (Wang et al. (2012)).

After the introduction of the `Word2Vec` model (Mikolov et al., 2013), numerous papers proposed derivations of the famous skip-gram algorithm for time annotated corpora (Frermann & Lapata, 2016). All these approaches attempt to acquire a better understanding language evolution by studying shifts in words semantic through time. Among them, Eger & Mehler (2016) learn linear temporal dependencies between word representations. Yao et al. (2018) learns diachronic word representations by matrix factorization with temporal alignment constraints. Bamler & Mandt (2017) proposed a temporal probabilistic skip-gram model with a diffusion prior. Rudolph & Blei (2017) also propose a probabilistic framework that uses exponential family embeddings. Compared to HMM and LDA based approached, the skip-gram algorithm uses standard gradient descent and can be parallelized easily to scale to massive corpora. But all the above approaches learn distinct word embeddings at each timestep, which leads a huge number of trainable parameters. An exception is Rosenfeld & Erk (2018) that combines a static word representation to a scalar timestep in a deep neural network that produces a temporal embedding.

An alternative to these various models is to leverage RNNs for language modeling. A recurrent language model takes a sequence of words of arbitrary size as input and outputs a probability distribution of the next word. Such models are often parameterized by LSTM networks (Hochreiter & Schmidhuber, 1997). Compared to the skip-gram algorithm that uses a limited context window, recurrent language models operate on sequences of arbitrary length, and can capture long-term dependencies. They are nowadays used at the core of an increasing number of tasks. For instance, as a feature extractor for text classification (Peters et al., 2018), as a core building block of unsupervised Neural Machine Translation models (Lample et al., 2018), or as a discriminator for Generative Adversarial Models on text (Yang et al., 2018).

In this paper, we propose a dynamic language model based on RNNs. The aim is to capture the language evolution through time via an end-to-end framework, where a standard RNN is conditioned by a latent representation of temporal drifts in language. To the best of our knowledge, no RNN LMs methods have been proposed for the extraction of temporal dynamics in text data.

3 MODEL

We propose a dynamic recurrent language network. It is a State Space Model (SSM) with one global latent state per timestep used to condition an LSTM Language Model. Unlike most current methods that learn complete word embedding matrices for each time step, we only learn one embedding per word, which is augmented with the states of the SSM. The LSTM can capture general language dynamics shared between all timesteps, and uses the temporal states to adapt language dynamics

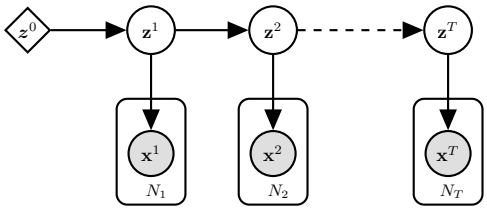

Figure 1: Directed graphical model.

depending on language bias specific to each timestep. We also learn a transition function between states that allows estimation of future states.

3.1 Notations and Task

We consider text sequences, annotated by discrete timesteps $t \in \{1, 2, \dots, T\}$, defined over a vocabulary of size V. Let $\mathbf{x}^t \in \mathbf{X}^t$ be a text sequence of length $|\mathbf{x}^t|$ published at time t. We denote by $\mathbf{x}_k^t$ the k^{th} term of sequence $\mathbf{x}^t$. A corpus $\mathbf{X}^t$ is composed of N_t such sequences. The complete corpus is denoted $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^T\}$.

In the standard, non-temporal, recurrent language modeling task, the objective is to find parameters $\boldsymbol{\theta}$ maximizing the likelihood of the next word given the previous ones for all sentences in a corpus:

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \prod_{\mathbf{x} \in \mathbf{X}} \prod_{k=1}^{|\mathbf{x}|-1} p_{\boldsymbol{\theta}}(\mathrm{x}_{k+1} | \mathbf{x}_{1:k}) \tag{1}$$

where $\mathbf{x}_{1:k}$ is the sequence of the first k words in the sequence $\mathbf{x}$ and $p_{\boldsymbol{\theta}}$ is parametrized by an RNN with parameters $\boldsymbol{\theta}$ that outputs next word probabilities. Specifically, we have $p_{\boldsymbol{\theta}}(\mathrm{x}_{k+1} | \mathbf{x}_{1:k}) = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}_k + \boldsymbol{b})$ where $\boldsymbol{W} \in \mathbb{R}^{V \times d_h}$ and $\boldsymbol{b} \in \mathbb{R}^V$ are parameters to learn, $\boldsymbol{h}_k = f(\mathrm{x}_k, \boldsymbol{h}_{k-1}; \boldsymbol{v})$ is a hidden vector of size d^h, and f is the RNN's recurrent function with parameters $\boldsymbol{v}$. We thus have $\boldsymbol{\theta} = \{\boldsymbol{U}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{v}\}$ where $\boldsymbol{U}$ is word embeddings matrix.

3.2 Dynamic Recurrent Language Model

Our goal is to extend classic recurrent language models with a dynamic component in order to adapt it to language shifts. We propose to augment the word embeddings of an LSTM LM with global temporal variables that are connected in time with a transition function learned jointly with the LSTM. We use a probabilistic SSM where global latent variables $\mathbf{z}^t \in \mathbb{R}^{d_z}$ capture information specific at a timestep t and onward, and are decoded by the LSTM to adapt its language model. Figure 1 represents the directed graphical model.

Learning the transition function between latent states enables estimation of future states of the system, where data is not available during training. The transition function is a Gaussian model centered on a non-linear transition function of the previous state, with a diagonal covariance matrix σ^2:

$$\mathbf{z}^{t+1} | \mathbf{z}^t \sim \mathcal{N}(g(\mathbf{z}^t; \boldsymbol{w}), \sigma^2)$$

where $\boldsymbol{w}$ are the transition function parameters. Learning a transition function gives the system the freedom to learn interesting latent trajectories. We hypothesize this will yield better performances, compared to a more restrictive diffusion model for instance. For the prior on the first timestep, we learn a vector $\boldsymbol{z}^0$ that acts as the initial conditions of the system.

Overall, the joint distribution factorizes as follows:

$$p_{\boldsymbol{\theta}, \boldsymbol{\psi}}(\mathbf{X}, \mathbf{Z}) = \prod_{t=0}^{T-1} p_{\boldsymbol{\psi}}(\mathbf{z}^{t+1} | \mathbf{z}^t) \prod_{t=1}^{T} \prod_{\mathbf{x} \in \mathbf{X}^t} p_{\boldsymbol{\theta}}(\mathbf{x} | \mathbf{z}^t) \tag{2}$$

where $\boldsymbol{\psi} = (\boldsymbol{w}, \sigma^2, \mathbf{z}^0)$ are the temporal prior parameters, and $\mathbf{Z} \in \mathbb{R}^{T \times d_z}$ is the matrix containing latent variables $\mathbf{z}^t$. $p_{\boldsymbol{\theta}}(\mathbf{x} | \mathbf{z}^t)$ is parameterized by an LSTM where the latent state $\mathbf{z}^t$ is concatenated to every word embedding vectors for a given sequence $\mathbf{x}$.

3.3 INFERENCE

Learning the generative model in equation 2 requires to infer the latent variables $\mathbf{z}^t$. In Bayesian inference, it is done by estimating their posterior $p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\mathbf{Z}|\mathbf{X}) = \frac{p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\mathbf{X},\mathbf{Z})}{\int p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\mathbf{X},\mathbf{Z})d\mathbf{Z}}$. Unfortunately, the marginalization on $\mathbf{Z}$ requires to compute an intractable normalizing integral. We therefore use Variational Inference (VI), and consider a variational distribution $q_{\boldsymbol{\phi}}(\mathbf{Z})$ that factorizes across all timesteps:

$$q_{\boldsymbol{\phi}}(\mathbf{Z}) = \prod_{t=1}^{T} q_{\boldsymbol{\phi}}^t(\mathbf{z}^t)$$

where $q_{\boldsymbol{\phi}}^t$ are independent Gaussian distributions $\mathcal{N}(\mu_t, \sigma_t^2)$ with diagonal covariance matrices σ_t^2, and $\boldsymbol{\phi}$ is the total set of variational parameters.

Following the derivation in Krishnan et al. (2017), we get the following evidence lower bound (ELBO):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}) = \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\phi}}^t}\left[\sum_{\mathbf{x}\in\mathbf{X}^t} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^t)\right] - \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\phi}}^{t-1}}\left[D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}^t(z^t)\|p_{\boldsymbol{\psi}}(z^t|z^{t-1}))\right] \quad (3)$$

with q^0 a Dirac distribution centered on $\boldsymbol{z}^0$, and D_{KL} the Kullback-Leibler (KL) divergence.

Since the observation model p_θ is an RNN, the model is non-conjugate, and the ELBO in equation 3 cannot be computed in closed form. We thus use the re-parametrization trick (Kingma & Welling, 2014; Rezende et al., 2014) to learn the model.

Global temporal states coupled with variational distributions independent in time offer several learning and computation advantages. This allows the system to deal with strong disruptions in language shifts, for which regularities observed on other steps could not hold. Rather than considerably upsetting the transition function, and thus highly impacting consecutive states, the learning algorithm can choose to ignore such difficult transitions, at a cost depending on the variance σ^2. This variance σ^2, learned jointly with the model, allows the learning algorithm to adapt the stochastic transition according to the regularity level of the data.

Moreover, since temporal dependency is broken, the computation of the KL in equation 3 can be computed in parallel, while still maintaining information flow through time. Indeed, since the latent states are global vectors, we can easily sample all of them at every optimization steps, even when T is large, and compute the prior in parallel. We can hence learn the model with mini-batches containing text samples from every timesteps, allowing gradient flow in $q_{\boldsymbol{\phi}}^t$ from the likelihood and the KL at both past and future timesteps simultaneously through the pathwise derivative.

4 EXPERIMENTAL PROTOCOL

We evaluate the proposed model together with baselines adapted from the temporal word embedding literature, detailed in section 4.1. We propose two evaluation configurations: prediction and modeling, presented in section 4.2. We perform experiments on three temporal corpora, with different sizes, temporal scales, and language level, described in section 4.3. In section 5.1 we conduct experiments on language modeling, and in section 5.2 we evaluate learned representations on classification tasks. In addition, we present text samples generated by our model in appendix D.

4.1 MODELS AND BASELINES

In our experiments, we compare the following models:

- **LSTM**: a standard regularized LSTM. This baseline has no temporal component, but is currently the state-of-the-art in language modeling.
- **DT**: the DiffDtime model presented in Rosenfeld & Erk (2018). It is a deep model that takes as input learned word embeddings and a timestep which outputs temporal word embeddings. Like our approach, this model learns only one embedding vector per word, but their temporal prior is obtained only by scaling a learned vector with a scalar timestep.

- **DWE**: the Dynamic Word Embedding model (Bamler & Mandt, 2017) learns Gaussian word embeddings with a probabilistic version of the skip-gram algorithm. This method learns a different set of word embeddings per timestep, that are smoothed in time with a diffusion prior.
- **DRLM-Id**: the Dynamic Recurrent Language Model proposed in this paper, where the transition function is replaced by the identity matrix so that $\mathbf{z}^{t+1} \sim \mathcal{N}(\mathbf{z}^t, \sigma^2)$.
- **DRLM**: the Dynamic Recurrent Language Model proposed in this paper with learned transition function.

Since, to the best of our knowledge, no dynamic recurrent language models have been proposed, we compare our approach to the distributed word embedding models DT and DWE. We adapt these models for recurrent language models by replacing the skip-gram component with an LSTM and discarding the context embeddings. More details can be found in appendix B. We also evaluate the proposed model without transition function (DRLM-Id) to assess its impact on performances.

4.2 Temporal Evaluation

To evaluate the models in a temporal context, we propose the two following settings:

Prediction We take the first T_p timesteps to train models, and evaluate them on the next timesteps $T_p + 1$ to T, with T the total number of timesteps. For DRLM, we use the transition model g to predict future states $\mathbf{z}^t$ in time. For DT and DWE we use the embeddings at the last training timestep $\mathbf{U}^{T_p}$. Timestep $T_p + 1$ is used for hyperparameters tuning.

Modeling In this configuration, models are trained and evaluated on all timesteps T, and corpora are randomly split into a training (60%) validation (10%) and test (30%) sets. In this setup, the training set for a given corpus is different from prediction, resulting in slightly different vocabulary sizes and hyperparameters.

We evaluate the models on language modeling and on downstream classification tasks. For language modeling the evaluation metric is perplexity on the respective test sets. We report micro perplexity, computed on the total test set, and macro perplexity, which is the averaged perplexity computed separately at each test timestep. For classification, we report precision, recall, and F1 measures for multi-label classification, and top1, top3, and top5 scores for multi-class classification.

4.3 Temporal Corpora

We use three different corpora for our experiment, which differ in terms of time ranges, topics, number of sequence per timestep, and sequence lengths.

- The **Semantic Scholar**[1] corpus (S2) is composed of titles from scientific papers published in machine learning conferences and journals from 1985 to 2017, split by years (33 timesteps). The corpus is composed of 50K titles, representing a total of 500K words.
- The **New York Times** (Yao et al., 2018) corpus (NYT) is composed of headlines from the New York Times newspaper spanning from 1990 to 2015, also split by years (26 timesteps). The corpus contains 50K headlines and 500K words.
- The **Reddit** corpus contains a sample of 3% of the social network's posts presented in Tan & Lee (2015). It is composed of 100K posts sampled from January 2006 to December 2013 split by quarters (32 timesteps).

4.4 Preprocessing and Hyperparameters

For each corpus, the vocabulary is constructed with words appearing at least 5 times in training sets (3 times for S2). The resulting vocabulary sizes are 5K tokens for S2, 8K for NYT and 13K for Reddit.

[1] `http://labs.semanticscholar.org/corpus/`

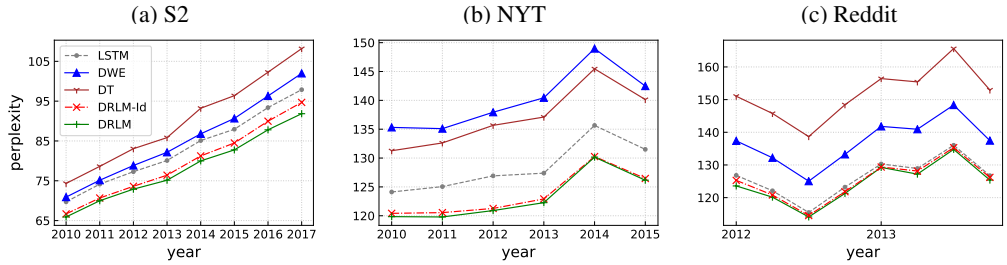

Figure 2: Perplexity through time for the *prediction* configuration. Results shown are obtained with texts published at future time periods, not seen during training.

We train a 2 layers AWD-LSTM (Merity et al., 2018b) for all models with hidden units and word embeddings of size 400. We use weight dropout, variational dropout, embedding dropout, and embeddings weight-tying (except for DWE that learns distinct word embeddings per timestep). Its hyperparameters are fixed across all models for a given corpus, except for input dropout, weight decay and learning rates that are tuned individually for each model.

For language modeling experiments, we decay the learning rate by a factor of 10 when no improvement is seen on the validation set for 10 consecutive epochs. For classification tasks, we linearly anneal the learning rate.

5 Results

In section 5.1 we present results on the temporal language tasks. In section 5.2, we extract the embeddings trained in section 5.1 and evaluate them on downstream classification tasks.

5.1 Language Modeling

We present here language modeling results for our three corpora in prediction and modeling configurations.

Prediction Figure 2 shows perplexity evolution for the prediction setup (numerical results are provided in appendix C). On the three corpora, both DRLM-Id and DRLM beat all baselines. The standard LSTM always performs better than the DWE and DT baselines that systematically overfit. This shows that LSTM, even without temporal components are powerful, and conditioning them is not trivial. Results on Reddit (figure 2c) tend to confirm this observation: performances LSTM, DRLM-Id, and DRLM are quasi-equivalent, with a gain of 2 points of micro perplexity for DRLM compared to LSTM. It is a corpus twice larger than the others, with longer sequences. Our analyses is that with sufficient data, and due to the auto-regressive nature of text, LSTM, even without explicit temporal prior, manage to capture temporal biases implicitly.

In the S2 corpus, we can see in figure 2a that, while the perplexity of DRLM-Id tends to converge to LSTM's perplexity, DRLM presents consistent improvement through time. On the NYT corpus, while DRLM-Id and DRLM have significant performance gain compared to LSTM (more than 5 points), the difference between the two models is small, and vanishes with time. The NYT corpus is composed of newspaper headlines that are greatly influenced by exterior factors, while S2 are scientific publications, which are influenced by one another through time. Hence DRLM, thanks to its transition function, is able to predict informative latent states on S2 but not on NYT.

Recursive Inference To validate empirically this hypothesis, we recursively infer the latent states $\mathbf{z}^t$ of DRLM by maximizing equation 3 with data from $\mathbf{X}^t$, and fixing all other parameters. We then evaluate the resulting model at $t+1$, next we infer $\mathbf{z}^{t+1}$ on $\mathbf{X}^{t+1}$, evaluate at $t+2$ and so on. We perform the same recursive inference algorithm for variational parameters of DWE. We call the two methods DRLM-F and DWE-F respectively, and present the results in figure 3.

We first observe that DRLM-F significantly improves long-term performances on NYT, meaning that the trained LSTM is able to interpret latent states $\mathbf{z}^t$ never seen during training. This is not

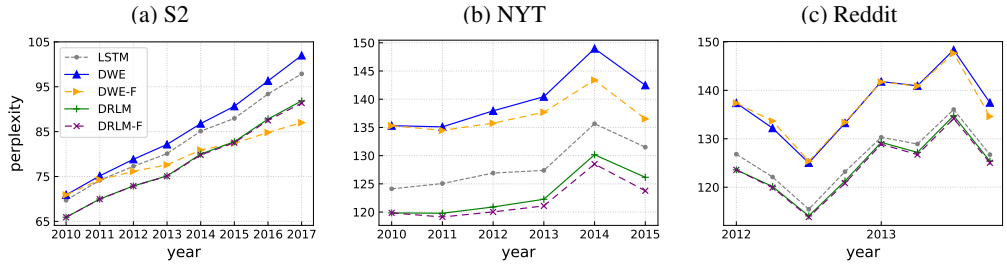

Figure 3: Perplexity through time with *recursive inference*. DRLM-F and DWE-F are trained on T_p timesteps, and then their variational parameters are recursively inferred on data at timestep $T_p + \tau$ and evaluated at $T_p + \tau + 1$. The LSTM baseline is displayed for comparison purposes.

Table 1: Modeling perplexity, where training and testing timesteps are the same.

	S2		NYT		Reddit	
Models	micro	macro	micro	macro	micro	macro
LSTM	62.8	66.2	109.9	110.4	116.7	123.0
DT	70.7	73.9	125.6	120.4	136.8	147.7
DWE	65.9	69.8	119.9	120.4	129.4	139.6
DRLM-Id	60.6	**61.3**	104.0	104.4	115.5	121.5
DRLM	**60.2**	**61.2**	**103.5**	**103.9**	**114.7**	**120.4**

trivial, given the difficulties and various tricks present in the literature to condition LSTM language models (Bowman et al., 2016; Semeniuta et al., 2017; Yang et al., 2017). We also see that recursive inference does not improve DRLM results on S2, while DWE results are greatly improved on NYT and even more on S2. This shows that the there is a temporal drift in S2, which is less clear on Reddit since recursive inference does not improve performances neither on DMLR nor on DWE. It then follows that DLRM predicts accurate latent states on the S2 corpus, since there is a temporal drift, but feeding future data does not enhance performances.

The DWE baseline benefits a lot more from recursive inference than DMLR. This is expected since it can adapt each word embedding at each timestep, whereas DRLM-F only infer the distribution of a single vector per timestep. This thus makes DWE-F a good baseline for assessing temporal drift.

Modeling Table figure 1 presents results for the modeling setup. As for prediction, temporal word embeddings baselines also fail to beat the LSTM baseline. All perplexities are lower since the task is easier, but DRLM and DRLM-Id keep their perplexity gain over LSTM. However, although DRLM is always at least better than DRLM-Id, the difference between the two is thinner than in the prediction setup.

5.2 Text Classification with Temporal Word Embeddings

To further evaluate the representations learned by DRLM, we extract its word embeddings augmented with temporal states, and use them for text classification. For the DT and DWE baselines, we learned temporal embeddings exactly as described in their respective papers.

Classification Model For every classification tasks, we learn a linear classifier that takes as input an average of the embeddings for a given sequence, as done in Joulin et al. (2016) and Shen et al. (2018). We learn the classifier by maximizing the likelihood $\frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N^t} y_i^t \log f(\boldsymbol{A}\boldsymbol{u}_i^t)$ where $\boldsymbol{A}$ is a weight matrix, $\boldsymbol{u}_i^t$ is the averaged temporal embeddings of sequence i at time t. And f is a normalizing function that depends on the task: a softmax for multi-class classification and a sigmoid for multi-label classification.

Tasks For S2, the task is to classify articles' keywords (multi-label with 400 classes). For NYT, the classes are the news section in which articles are published (mono-label with 28 labels). And for Reddit, the task is to classify the subreddit in which posts are submitted (mono-label with 60 labels).

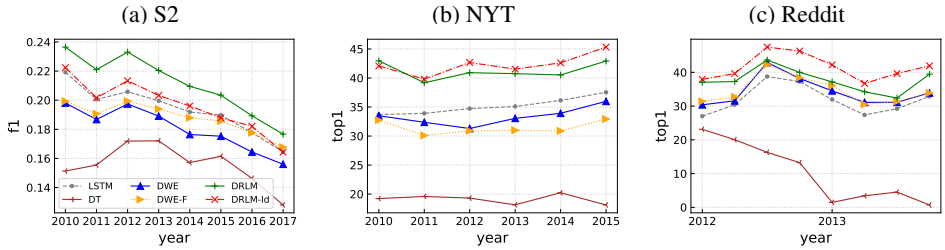

Figure 4: Classification results with temporal word embeddings in the *prediction* configuration. For LSTM, DRLM, and DRLM-Id, word embeddings were pretrained on the language modeling tasks in section 5.1, while for the baselines DT and DWE, they were trained as proposed by their authors.

Table 2: Classification results with temporal word embeddings for the *modeling* configuration.

	S2			NYT			Reddit		
Models	Precision	Recall	F1	top1	top3	top5	top1	top3	top5
LSTM	0.427	0.171	0.221	41.4	63.2	74.3	44.0	66.7	74.8
DT	0.272	0.077	0.110	17.3	40.9	57.5	40.9	56.4	63.3
DWE	0.371	0.129	0.174	24.8	51.0	66.9	44.5	64.0	71.5
DRLM-Id	0.429	**0.176**	**0.226**	44.3	68.2	79.5	**45.6**	**68.6**	**76.7**
DRLM	**0.435**	**0.177**	**0.227**	**44.8**	**68.8**	**80.0**	45.2	**68.7**	**76.8**

Results Prediction results are displayed in figure 4, and the detailed results can be found in appendix C. For the DWE, we also show results when the word embeddings are recursively inferred with the skip-gram filtering algorithm of Bamler & Mandt (2017) (DWE-F). Modeling results are given in table 2.

Classification results confirm language modeling results. On the S2 corpus, we can see in figure 4a that DRLM has higher F1 score across all timesteps, and performs consistently better in the modeling configuration. However, DRLM-Id is better in prediction for NYT (figure 4b) except in 2010, which is the validation timestep. And for the same configuration, DWE also performs better than DWE-F, confirming chaotic temporal drifts in the corpus. Reddit is the only corpus where a temporal word embeddings baseline, DWE, manages to beat the LSTM. Overall, our DLMR and DLMR-Id models obtained significantly better classification scores compared to every baselines.

6 Conclusion

We proposed a Dynamic Recurrent Language Model (DRLM) for handling temporal drifts in language. We conditioned and LSTM language model by augmenting its word embeddings with temporal latent variables. The latent variables are global states that capture temporal variations. They are structured in time with a learned transition model, which enables the estimation of future states. The transition model acts as a temporal prior on latent states, allowing their distribution to adapt to disruptive shifts in the data.

Experiments on three corpora with various sizes, time scales, and language levels, showed that our approach beats temporal word embeddings baselines in two temporal evaluation configurations (prediction and modeling) on language modeling and classification tasks. Empirical results also showed that, for certain corpora, our model is able to accurately predict future states at timesteps where no data were available during training.

In this work, we did not address the fact that new words appear at future timesteps. Handling out-of-vocabulary words are of major concern in NLP, and predicting new words is even more challenging. A promising approach would be to use byte pair encoding (Sennrich et al., 2016) and learn to predict future sub-words combinations.

Also, while our method works well in practice, it is not clear how the LSTM networks internals interpret the latent states. Efficiently conditioning language models is an active research field, as stated in section 5.1. And since convolutional decoders' popularity is rapidly growing (Semeniuta

et al., 2017; Yang et al., 2017), using efficient methods from the computer vision community, like Adaptive Instance Normalization (Huang & Belongie, 2017), seems a promising research direction.

REFERENCES

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Robert Bamler and Stephan Mandt. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 380–389, 2017.

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

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. *arXiv preprint arXiv:1712.01769*, 2017.

Steffen Eger and Alexander Mehler. On the linearity of semantic change: Investigating meaning variation via dynamic graph models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016.

William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the _. In *Proceedings of the International Conference on Learning Representations*, 2018.

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

William L Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 1489–1501, 2016.

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

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 328–339, 2018.

Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pp. 1510–1519, 2017.

Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Sequential modeling of topic dynamics with multiple timescales. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):19, 2012.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

Ata Kabán and Mark A Girolami. A dynamic probabilistic model to visualise topic evolution in text streams. *Journal of Intelligent Information Systems*, 18(2-3):107–125, 2002.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pp. 61–65, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.

Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *AAAI*, pp. 2101–2109, 2017.

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

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*, 2018.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *Proceedings of the International Conference on Learning Representations*, 2018.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. An analysis of neural language modeling at multiple scales. *arXiv preprint arXiv:1803.08240*, 2018a.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. In *Proceedings of the International Conference on Learning Representations*, 2018b.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

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

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

Alex Rosenfeld and Katrin Erk. Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 474–484, 2018.

Maja Rudolph and David Blei. Dynamic bernoulli embeddings for language evolution. *arXiv preprint arXiv:1703.08052*, 2017.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 627–637, 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 1715–1725, 2016.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*, 2018.

Chenhao Tan and Lillian Lee. All who wander: On the prevalence and characteristics of multi-community engagement. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1056–1066. International World Wide Web Conferences Steering Committee, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2017.

Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.

Xuerui Wang and Andrew McCallum. 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, 2006.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3881–3890, 2017.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. *arXiv preprint arXiv:1805.11749*, 2018.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, pp. 673–681, 2018.

A COMPLETE INFERENCE DERIVATION

The ELBO in equation 3 can be obtained as follows:

$$\begin{aligned}
\log P_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{X}) &= \log \int_{\mathbf{Z}} p_{\boldsymbol{\psi}}(\mathbf{Z}) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{X}^t|\mathbf{z}^t) d\mathbf{Z} \\
&= \log \int_{\mathbf{Z}} q_{\boldsymbol{\phi}}(\mathbf{Z}) p_{\boldsymbol{\psi}}(\mathbf{Z}) \frac{\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{X}^t|\mathbf{z}^t)}{q_{\boldsymbol{\phi}}(\mathbf{Z})} d\mathbf{Z} \\
&\geq \int_{\mathbf{Z}} q_{\boldsymbol{\phi}}(\mathbf{Z}) \log \left(p_{\boldsymbol{\psi}}(\mathbf{Z}) \frac{\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{X}^t|\mathbf{z}^t)}{q_{\boldsymbol{\phi}}(\mathbf{Z})} \right) d\mathbf{Z} \\
&= \sum_{t=1}^{T} \int_{\mathbf{z}^t} q^t_{\boldsymbol{\phi}}(\mathbf{z}^t) \log p_{\boldsymbol{\theta}}(\mathbf{X}^t|\mathbf{z}^t) d\mathbf{z}^t \\
&+ \sum_{t=1}^{T} \int_{\mathbf{z}^{t-1}} q^{t-1}_{\boldsymbol{\phi}}(\mathbf{z}^{t-1}) \int_{\mathbf{z}^t} q^t_{\boldsymbol{\phi}}(\mathbf{z}^t) \log \frac{p_{\boldsymbol{\psi}}(\mathbf{z}^t|\mathbf{z}^{t-1})}{q^t_{\boldsymbol{\phi}}(\mathbf{z}^t)} d\mathbf{z}^{t-1} d\mathbf{z}^t \\
&= \sum_{t=1}^{T} \mathbb{E}_{q^t_{\boldsymbol{\phi}}(\mathbf{z}^t)} \left[\log p_{\boldsymbol{\theta}}(\mathbf{X}^t|\mathbf{z}^t) \right] - \sum_{t=1}^{T} \mathbb{E}_{q^{t-1}_{\boldsymbol{\phi}}(\mathbf{z}^{t-1})} \left[D_{\mathrm{KL}}(q^t_{\boldsymbol{\phi}}(\mathbf{z}^t) \| p_{\boldsymbol{\psi}}(\mathbf{z}^t|\mathbf{z}^{t-1})) \right] \\
&= \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi})
\end{aligned}$$

where the inequality is obtained thanks to the Jensen theorem on concave functions.

The KL between two Gaussians owns an analytically closed form. This allows us to rewrite our log-likelihood lower-bound, noted $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi})$, as follows:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}) &= \sum_{t=1}^{T} \mathbb{E}_{q^t_{\boldsymbol{\phi}}(\mathbf{z}^t)} \left[p_{\boldsymbol{\theta}}(\mathbf{X}^t|\mathbf{z}^t) \right] + \frac{Td}{2} \\
&\quad - \frac{1}{2} \left(T \sum_{i=0}^{d-1} \log \sigma_i^2 - \sum_{t=1}^{T} \sum_{i=0}^{d-1} \log \eta_{t,i}^2 + \sum_{t=1}^{T} \sum_{i=0}^{d-1} \frac{\eta_{t,i}^2}{\sigma_i^2} \right. \\
&\quad \left. + \sum_{t=1}^{T} \mathbb{E}_{q^{t-1}_{\boldsymbol{\phi}}(\mathbf{z}^{t-1})} \left[(g(\mathbf{z}^{t-1}; \boldsymbol{w}) - \mu_t)' (\sigma^2)^{-1} (g(\mathbf{z}^{t-1}; \boldsymbol{w}) - \mu_t) \right] \right)
\end{aligned}$$

where we note A' the matrix transpose of a matrix A and where σ_i^2 and $\eta_{t,i}^2$ stand for the i-th component of diagonals σ^2 and η_t^2 respectively. This re-writing allows one to improve learning stability w.r.t. a version in which sampling would be done on the KL components too.

B DERIVING TEMPORAL WORD EMBEDDING METHODS FOR RECURRENT LANGUAGE MODELING

We detail here how we adapt temporal word embeddings baselines to recurrent language modeling. The baselines are Dynamic Word Embeddings (DWE) (Bamler & Mandt, 2017), and DiffTime (Rosenfeld & Erk, 2018). For both methods, we get rid of the context embeddings and only keep word embeddings $\mathbf{U}$.

B.1 DYNAMIC WORD EMBEDDINGS

In DWE (Bamler & Mandt, 2017), Gaussian word embeddings are learned at each timestep with a temporal diffusion prior:

$$\mathbf{U}_{t+1} | \mathbf{U}_t \sim \mathcal{N} \left(\frac{U_t}{1 + \sigma_t^2 / \sigma_0^2}, \frac{1}{\sigma_t^{-2} + \sigma_0^{-2}} I \right)$$

where σ_0^2 and σ_t^2 are hyperparameters of the model.

We derive their skip-gram algorithm for our setting by maximizing the following approximate ELBO:

$$\begin{aligned}\mathcal{L}_{\mathcal{DWE}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = &\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{U}_t)} \left[\log p_{\boldsymbol{\theta}}(\mathbf{X}^t|\mathbf{U}^t)\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{U}_t)} \left[\log \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{U}_{t-1})} \left[p(\mathbf{U}_t|\mathbf{U}_{t-1})\right]\right] \\ &- \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{U}_t)} \left[\log q_{\boldsymbol{\phi}}(\mathbf{U}_t)\right]\end{aligned} \tag{4}$$

where $p_{\boldsymbol{\theta}}$ is paramatrized by an LSTM. $q_{\boldsymbol{\phi}}$ is a variational Gaussian distribution that factorizes as:

$$q_{\boldsymbol{\phi}}(\mathbf{U}) = \prod_{t=1}^{T} q_{\boldsymbol{\phi}}(\mathbf{U}_t)$$

and $\boldsymbol{\phi}$ are its parameters.

To learn this model, we sample a mini-batches $\mathbf{M}$ that contains text coming from different training timesteps. We must hence rescale the ELBO in equation 4. We do so by estimating the probability that a given word appears in a particular mini-batch:

$$\begin{aligned}\mathcal{L}_{minibatch}(\boldsymbol{\theta}, \boldsymbol{\phi}) = &\frac{|\mathbf{X}|}{|\mathbf{M}|} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{U}^{\mathbf{M}})} \left[\sum_{\mathbf{x} \in \mathbf{M}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{U}^{\mathbf{M}})\right] \\ &+ \sum_{\mathbf{u} \in \mathbf{U}^{\mathbf{M}}} \frac{1}{(1-(1-\nu_{\mathbf{u}})^{|\mathbf{M}|})} \sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{u})} \left[\log \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{u}_{t-1})} \left[p(\mathbf{u}_t|\mathbf{u}_{t-1})\right]\right] \\ &- \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{u}_t)} \left[\log q_{\boldsymbol{\phi}}(\mathbf{u}_t)\right]\end{aligned}$$

where $\mathbf{U}^{\mathbf{M}}$ are the embeddings of words in $\mathbf{M}$, $\nu_{\mathbf{u}}$ is the apparition frequency of term whose embedding is $\mathbf{u}$ in $\mathbf{X}$, and $|\mathbf{X}|$ (respectively $|\mathbf{M}|$) is the number of words in $\mathbf{X}$ ($\mathbf{M}$). In this formulation, gradient computation does not require any approximation, while allowing it to flow through all timesteps.

B.2 DIFFTIME

The adaptation of the DiffTime baseline (Rosenfeld & Erk, 2018) is straightforward. It learns a non-linear function d that outputs temporal word embeddings:

$$\mathbf{u}_t = d(\mathbf{u}, t; \boldsymbol{\phi})$$

where $\mathbf{u}$ is a learned word embedding, t is a scalar timestep, and $\boldsymbol{\phi}$ are the function's parameters. We refer the reader to the complete paper for more details on the implementation of d.

For recurrent language modeling adaptation, we simply learn jointly the word embeddings $\mathbf{U}$, the parameters $\boldsymbol{\phi}$ of d and the parameters $\boldsymbol{\theta}$ of an LSTM by maximizing the following likelihood:

$$\mathcal{L}_{\mathcal{DT}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{U}) = \prod_{t=1}^{T} \prod_{\mathbf{x} \in \mathbf{X}^t} \prod_{k=1}^{|\mathbf{x}|-1} p_{\boldsymbol{\theta}}(\mathrm{x}_{k+1}|\mathbf{u}_{1:k}^t)$$

C QUANTITATIVE RESULTS FOR PREDICTION

We report here the quantitative results for the prediction configuration. Language modeling results are shown in table 3, and text classification results in table 4.

D TEXT GENERATION THROUGH TIME

We present here texts samples generated by our model. The three word triplets that most often appear in the S2 modeling test set are used as seed for LSTM. Samples are generated by beam search with the DLMR model trained in modeling configuration. Table 5 presents generated samples where the latent state that condition the LSTM evolves from $\mathbf{z}^0$ to $\mathbf{z}^T$.

Table 3: Prediction perplexity

	S2		NYT		Reddit	
Models	micro	macro	micro	macro	micro	macro
LSTM	84.7	82.7	128.5	128.4	125.8	126.1
DT	92.0	89.6	137.1	137.0	151.1	151.6
DWE	87.0	84.8	140.1	140.0	136.5	139.9
DRLM-Id	81.2	79.2	123.7	123.6	124.7	125.0
DRLM	**79.7**	**77.8**	**123.3**	**123.1**	**123.9**	**124.3**

Table 4: Prediction classification

	S2			NYT			Reddit		
Models	Precision	Recall	F1	top1	top3	top5	top1	top3	top5
LSTM	0.378	0.144	0.190	35.1	54.7	65.9	32.0	52.8	63.4
DT	0.357	0.111	0.154	19.1	45.1	62.1	12.5	26.0	33.2
DWE	0.358	0.134	0.177	33.4	55.0	66.8	34.3	51.0	60.8
DWE-F	0.317	**0.166**	0.186	31.4	53.4	65.0	34.8	50.7	59.7
DRLM-Id	**0.385**	0.144	0.193	**42.3**	**66.9**	**77.6**	**41.6**	**58.3**	**67.0**
DRLM	0.370	**0.167**	**0.208**	41.2	60.0	68.9	38.0	56.2	66.7

Table 5: Text sequences generating with DMLR conditioned on different timesteps on the S2 corpus. The first three words are uses as seeds, and the samples are generted by beam search.

1985	a framework for shape recovery from images
1995	a framework for shape recovery from images
2000	a framework for automatic evaluation of machine translation experiments
2005	a framework for automatic evaluation of statistical machine translation
2010	a framework for unsupervised learning of named entity recognizers
2015	a framework for unsupervised feature selection
2016	a framework for unsupervised learning of deep neural networks
2017	a framework for training deep convolutional neural networks

---

1985	unsupervised learning of hidden markov models
1990	unsupervised learning of hidden markov models
1995	unsupervised learning of gaussian graphical models
2000	unsupervised learning of gaussian graphical models
2005	unsupervised learning of named entity recognizers
2010	unsupervised learning of gaussian graphical models
2015	unsupervised learning of deep convolutional neural networks
2016	unsupervised learning of convolutional neural networks
2017	unsupervised learning of generative adversarial networks

---

1985	a comparison of smoothing techniques for statistical machine translation
1990	a comparison of smoothing techniques for statistical machine translation
1995	a comparison of smoothing techniques for word sense disambiguation
2000	a comparison of smoothing techniques for word sense disambiguation
2005	a comparison of smoothing techniques for statistical machine translation
2010	a comparison of smoothing techniques for statistical machine translation
2015	a comparison of convolutional neural networks for action recognition
2016	a comparison of convolutional neural networks for action recognition
2017	a comparison of convolutional neural networks for action recognition