Synthetic Text Generation for Sentiment Analysis

Umar Maqsud
Technische Universität Berlin
Straße des 17. Juni 135, 10623 Berlin, Germany
umar.maqsud@campus.tu-berlin.de

Abstract

Natural language is a common type of input for data processing systems. Therefore, it is often required to have a large testing data set of this type. In this context, the task to automatically generate natural language texts, which maintain the properties of real texts is desirable. However, current synthetic data generators do not capture natural language text data sufficiently. In this paper, we present a preliminary study on different generative models for text generation, which maintain specific properties of natural language text, i.e., the sentiment of a review text. In a series of experiments using different data sets and sentiment analysis methods, we show that generative models can generate texts with a specific sentiment and that hidden Markov model based text generation achieves less accuracy than Markov chain based text generation, but can generate a higher number of distinct texts.

1 Introduction

Text generation is the task of automatically generating texts, which maintain specific properties of real texts. In the context of synthetic text generation, generative models are used to generate test data for benchmarking big data systems (Rabl and Jacobsen, 2012). BDGS (Ming et al., 2014) is a text generator that applies latent dirichlet allocation (Blei et al., 2003) as the text data generation model and BigBench (Ghazal et al., 2013) is a benchmark that provides a text generator based on Markov chain model (Rabiner, 1989).

Sentiment analysis (SA) is a method of processing opinions and subjectivity of a text. The task is to find and extract the sentiment polarity expressed in a text.

The goal of the paper is to demonstrate the ability of different generative models, i.e., latent dirichlet allocation (LDA), Markov chains (MC), and hidden Markov model (HMM), to generate text with a specific sentiment. This is an important problem because the sentiment of a text may be crucial in several applications like extracting the customers reviews about books, movies, or food and classifying them along their sentiment.

The contributions of this paper are as follows. We present a primary study on three different generative models for text generation. LDA and MC are used for text generation in previous work (Ming et al., 2014; Ghazal et al., 2013). We introduce the well known HMM to use it for text generation and compare it with LDA and MC. In a series of experiments, we analyze the scalability, cardinality, and the ability to generate text with a sentiment. For sentiment analysis, we use state-of-the-art methods. The evaluation indicates that the models can generate texts with a specific sentiment. The hidden Markov model achieves a lower accuracy than Markov chains, but can generate more distinct texts.

The remainder of the paper is organized as follows. Sections 2 and 3 provide an overview on generative models and sentiment analysis approaches. In Section 4 the results of the preliminary experiments are presented. Finally, Section 5 presents a summary and discusses directions for future work.

2 Generative Models

We describe in this section the previously mentioned generative models for text generation.

2.1 Latent Dirichlet Allocation

Latent dirichlet allocation (LDA) is a generative probabilistic model and can be applied for text generation (Ming et al., 2014). Documents are modeled as mixtures over latent topics and topics

Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2015), pages 156–161,
Lisboa, Portugal, 17 September, 2015.

are described by a distribution over words. The generation process in LDA has following steps for each document, as described in (Blei et al., 2003):

1. Choose $N \sim$ Poisson(ξ) as the length of a the document.
2. Choose $\theta \sim$ Dir(α) as the mixture of latent topics of the document.
3. For each of N words w_n:
   (a) Choose a topic $z_n \sim$ Multinomial(θ)
   (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

To learn a LDA model of text documents the library lda-c[1] is used. BDGS (Ming et al., 2014) is used to generate text based on these models.

2.2 Markov Chain

A Markov chain is a sequence of random variables with the Markov property (Rabiner, 1989). Suppose $X = (X_1, X_2...X_T)$ is a sequence of random variables and $W = (w_1, w_2 \dots w_n)$ the state space. Then the Markov property is:

1. Transition probability depends only on the previous state.
   $P(X_t = w_i|X_1, \dots, X_{t-1}) = P(X_t = w_i|X_{t-1})$
2. Transition probability depends on k previous states (k-order markov chain).
   $P(X_t = w_i|X_1, \dots, X_{t-1}) = P(X_t = w_i|X_{t-k}, \dots, X_{t-1})$

A first order Markov chain will only consider the probability of a word appearing after another one. To get more realistic text, higher order n-gram models should be used for generating the text (Ghazal et al., 2013).

2.3 Hidden Markov Model

A hidden Markov model (HMM) is a Markov process with unobserved states and an observable variable (Rabiner, 1989). The hidden states have a probability distribution over the possible observable outputs. Suppose $X = (X_1, X_2 \dots X_T)$ is a sequence of hidden random variables, $H = (h_1, h_2 \dots h_n)$ the state space and $O = (o_1, o_2, \dots o_m)$ an observable variable. Additionally to MC, HMM is defined as:

1. Observation probability depends only on the current hidden state.
   $P(O_t = o_j|X_t = h_i)$

A basic task of HMM is the supervised learning process, where given a set of hidden and observed sequences, the most likely model that produced the observed sequence is searched. A typical application for this problem is part-of-speech tagging, where the observed variables are the words and the hidden states are the part-of-speech tags (Brants, 2000; Cutting et al., 1992).

HMM is used for text generation as follows. First, the text is tagged using a part-of-speech tagger from the Stanford CoreNLP library (Manning et al., 2014). Then, the most likely model that produces those sequences is computed. The hidden state transitions and observations are counted and used as relative frequencies to estimate the transition probabilities.

3 Sentiment Analysis

Two different approaches of sentiment analysis can be identified. The first approach uses lexicons to retrieve the sentiment polarity of a text. This lexicons contain dictionaries of positive, negative, and neutral words and the sentiment polarity is retrieved according to the words in a text. Machine learning uses annotated texts with a given sentiment to build a classification model. Sentiment analysis is implemented as a binary classification problem (Pang et al., 2002).

3.1 SentiWordNet

SentiWordNet (Baccianella et al., 2010) is a widely used lexical resource in sentiment analysis and is based on the English lexical dictionary WordNet (Miller, 1995). This lexical dictionary groups words into synonym sets, which are called synsets, and provides relations between these synsets. SentiWordNet associates each synset with three numeric polarity scores: *positive*, *negative* and *neutral*.

To retrieve the sentiment of a word based on this lexicon, the average scores of all associated synsets of a given word are considered and it is assessed as to be positive, if the average score of the

[1] http://www.cs.princeton.edu/~blei/lda-c

positive polarity is greater than that of the negative. The overall average of all words is calculated to assess the sentiment of a text.

3.2 Supervised Classification

Machine learning can be applied to build a supervised classification model. Text elements are represented by a feature vectors. The features can be the words of the text or their part-of-speech tags.

Support vector machines (SVMs) have been shown to be appropriate for text categorization (Joachims, 1998). In binary classification, the task is to find a hyperplane that separates the document vectors in the two classes and to maximize the margin between them. SVMs are widely used in sentiment analysis (Pang et al., 2002).

For training and testing LibShortText library[2] is used (Fan et al., 2008).

3.3 Stanford Sentiment Treebank

Socher et al. (2013) have introduced a treebank, which includes phrases and sentences annotated with fine-grained sentiment labels. In the five class fine-grained classification task following labels are used: *very negative*, *negative*, *neutral*, *positive*, and *very positive*.

As described in (Manning et al., 2014), sentiment analysis is performed with a model over parse trees. Nodes of a parse tree of each sentence are given a sentiment score. The overall score of the sentence is given at the root node of the parse tree. But it is unclear how to combine the sentiments over many sentences. We count all sentiment representations and take the mean as the overall sentiment of a set of sentences.

4 Experiments

In a series of experiments we analyzed the scalability, cardinality and the ability to generate text with a sentiment.

4.1 Experiment 1: Scalability

In this experiment the scalability of the presented models are measured on data sets of different sizes.

We use the food reviews data set used in (McAuley and Leskovec, 2013) and construct seven sub data sets with 10K, 50K, 100K, 200K, 300K and 500K food reviews respectively. We measure the execution time of the learning algorithms of the models on each of these sub data sets.

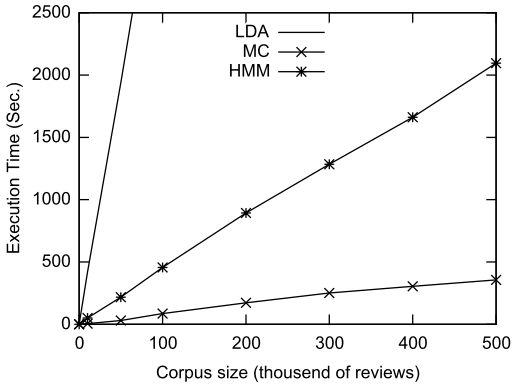

Figure 1: Execution time of LDA, MC and HMM on data sets of different sizes. HMM achieves a smaller execution time than LDA but greater than MC.

Figure 1 shows for each sub data set the execution time of the learning phase. As we can see, MC outperforms the other methods in terms of scalability because it only builds n-grams. HMM has a higher execution time because the data sets have to be tagged using a part-of-speech tagger. LDA performs the worst due to the extensive learning phase.

4.2 Experiment 2: Cardinality

In this experiment the cardinality of the synthetic data sets are measured. The cardinality is defined as the amount of distinct text elements in the generated data set. Two text elements are the same if they have the exact same string. A text element can be an arbitrary type of text, i.e. a sentence or a document. This will show the upscaling behavior in terms of the ability to generate distinct texts.

We use a data set of 10,662 movie reviews used in (Pang and Lee, 2005), which contains an equal number of positive and negative reviews, and divide it into two data sets along their sentiment polarity. On both data sets we build the presented models, which we utilize to scale up by factors of 1, 2, 10, 100 and 1000.

Figure 2 shows that the LDA and HMM models performs best in generating distinct text elements, where almost all text elements are distinct. The MC model generates the smallest amount of distinct text elements, e.g. only 62% distinct text elements using scale up factor 1000. The next word in LDA and HMM only depend on the latent vari-

[2] http://www.csie.ntu.edu.tw/~cjlin/libshorttext/

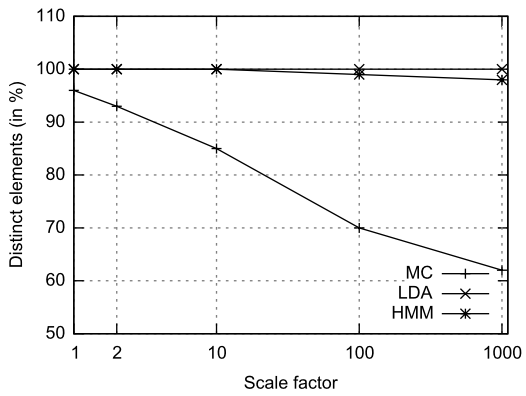

Figure 2: The relative amount of distinct text elements in the synthetic datasets. The synthetic data sets are generated by scale up factors of 1, 2, 10, 100 and 1000. MC generates the smallest amount of distinct text elements, while LDA and HMM generate almost no duplicates.

	positive			negative		
	(a)	**(b)**	**(c)**	**(a)**	**(b)**	**(c)**
Original	63	75	**79**	57	75	**79**
LDA (1x)	60	73	68	52	71	58
LDA (2x)	62	70	68	52	69	59
LDA (10x)	63	70	69	55	67	59
MC (1x)	62	72	75	54	72	70
MC (2x)	62	73	75	55	**73**	72
MC (10x)	63	74	**76**	56	**73**	72
HMM (1x)	61	69	73	54	68	68
HMM (2x)	61	71	73	54	70	67
HMM (10x)	62	71	73	54	70	67

Table 1: This table shows the F-measures of the original and synthetic data sets for the positive and negative class separately. The synthetic data sets are generated by scale up factors of 1, 2 and 10. The sentiments analysis methods are SentiWordNet (a) SVM (b), and Stanford library (c). The HMM achieves a lower F-measure than MC but a higher than LDA on each scale up factor.

able and not on the previous words, where in LDA it depends on the latent topics and in HMM on the part-of-speech tags. Therefore, more combinations of words are possible.

4.3 Experiment 3: Sentiment-aware Text Generation

In this experiment it is demonstrated that the models learn high-quality language presented models and are able to generate text with a sentiment.

We use the same data set as in the previous experiment and divide it into two data sets along their sentiment polarity. To build an SVM based classifier we split each data set into a training and test data set. On both data sets we learn the presented models, which we utilize to scale up by factors of 1, 2 and 10. We use (a) SentiWordNet (Baccianella et al., 2010), (b) SVM, and (c) the Stanford sentiment analysis library (Socher et al., 2013) to assess whether the generated reviews have the appropriate sentiment.

Table 1 shows the main experimental results. We see that the HMM is more accurately than LDA but less accurately than the MC. The method (c) outperforms the other methods and achieves an F-measure of 79% for the positive and 79% for the negative class. The basic methods (a) and (b) reveal only a modest difference between the original and synthetic data set, while the advanced method (c) illustrates a significant decrease of the F-measure in the synthetic data sets. One reason why the F-measure have declined is that basic methods work by assessing words in isolation. They give positive scores for positive words and negative scores for negative words and then aggregate these scores. Therefore, the order of words is ignored. In contrast, the advanced method builds a representation of the whole sentence based on the sentence structure using the parse tree. Consequently, MC and HMM perform better than LDA because of their ability to capture the order of words.

The F-measures of all models and sentiment analysis methods are almost constant on each scale up factor, which indicates a robust upscaling behavior of these models. The HMM achieves a lower F-measure than MC, but can generate a higher number of distinct text elements than MC.

Figure 3 shows the sentiment polarity of the original data set and synthetic data sets. The first column is the original data set tagged by the Stanford library and is classified about 40% as positive, 49% as negative and 11% as neutral. As we can see, the sentiment polarity of the synthetic data set using MC is most similar to the original one, with about 36% tagged as positive, 43% as negative and 21% as neutral. The experiments indicate that the presented models can generate texts with a specific sentiment.

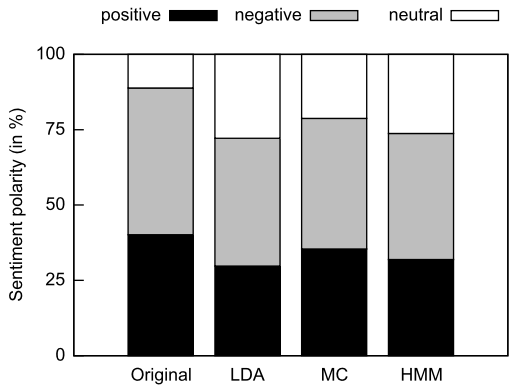

Figure 3: This figure shows the sentiment polarity of the original data set and synthetic data sets. For sentiment analysis the Stanford library is used. The sentiment polarity of the synthetic data set using MC is most similar to the original one.

4.4 Samples from the models

In this section we qualitatively investigate the capabilities of the presented models. The simplest qualitative experiment is to inspect the samples generated by the three models. We use the movie reviews data set and divide it into two data sets along their sentiment polarity. On both data sets we build the presented models, which we utilize to generate the samples.

The samples below were obtained by generating random texts 10 times and selecting the most interesting sample. The samples declared as *negative sentiment*, for example, are retrieved from a model learned on the negative sub data set.

4.4.1 Samples from the LDA model

negative sentiment:

credits i'll emotional uglier camera and can see moore's wanes reason film written to is by her that that that rather

positive sentiment:

simple interdependence particularly and quinn with baran rich questing delicate messenger on wallet comedy-drama such funny check a , . .

4.4.2 Samples from the MC model

negative sentiment:

a little thin , leaving these actors , that it gave me no reason to see the same illogical things keep happening over and over again .

positive sentiment:

often shocking but ultimately worthwhile exploration of the acting , have made to be introverted young men set out to be viewed and treasured for its straight-ahead approach to visualizing nijinsky's diaries is both inspiring and pure of heart , you can't go home again

4.4.3 Samples from the HMM model

negative sentiment:

in his franchise , chou-chou , " the exercise at the love ah-nuld attempted than drama , but pretty predictably , this splitting of the plays to funny routines title of there 's badly the director , and no beautiful life which is someone on a stagy episode .

positive sentiment:

you is hard n't of beautiful updating comedy complex family – be acquainted the usual recipe at every quiet but laughs truly a melodramatic at the in her wholesome , heartwarming david that 's an inevitable bio-pic with museum .

5 Conclusion

In this paper, we presented a primary study on generative models for text generation. A series of experiments indicate that the presented models can generate texts with a specific sentiment. The hidden Markov model achieves a lower F-measure than Markov chain, but can generate a higher number of distinct texts than Markov chains.

In future evaluations the methods will be analyzed within larger and different data sets. Future work will also investigate other generative models for text generation. Grave et al. (2014) introduced a generative model of sentences with latent variables, which takes the syntax into account by using syntactic dependency trees. Sutskever et al. (2011) uses recurrent neural networks to build statistical language models, which can be utilized to generate text.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Thorsten Brants. 2000. Tnt – A statistical part-of-speech tagger. In *ANLP*, pages 224–231.

Douglas R. Cutting, Julian Kupiec, Jan O. Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In *ANLP*, pages 133–140.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. 2013. Bigbench: towards an industry standard benchmark for big data analytics. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 1197–1208.

Edouard Grave, Guillaume Obozinski, and Francis R. Bach. 2014. A markovian approach to distributional semantics with application to semantic compositionality. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1447–1456.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK. Springer-Verlag.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Julian J. McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. *CoRR*, abs/1303.4402.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Zijian Ming, Chunjie Luo, Wanling Gao, Rui Han, Qiang Yang, Lei Wang, and Jianfeng Zhan. 2014. BDGS: A scalable big data generator suite in big data benchmarking. *CoRR*, abs/1401.5465.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Lawrence R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb.

Tilmann Rabl and Hans-Arno Jacobsen. 2012. Big data generation. In *Specifying Big Data Benchmarks - First Workshop, WBDB 2012, San Jose, CA, USA, May 8-9, 2012, and Second Workshop, WBDB 2012, Pune, India, December 17-18, 2012, Revised Selected Papers*, pages 20–27.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October. Association for Computational Linguistics.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1017–1024.