SAM: Semantic Attribute Modulated Language Modeling

Wenbo Hu[1], Lifeng Hua[2], Lei Li[2], Tian Wang[3], Jun Zhu[1], Hang Su[1], Bo Zhang[1]
[1] 1Dept. of Comp. Sci. & Tech. Tsinghua University
[2] Toutiao Lab, [3] eBay
{hwb13@mails., dcszj@, suhangss@, dcszb@}tsinghua.edu.cn;
{lileilab, hualifeng}@bytedance.com; twang5@ebay.com

July 4, 2017

Abstract

As a fundamental task in natural language processing, language modeling aims to estimate the distribution of the word sequences. However, most existing algorithms have focused on the main text while often ignoring the vastly-accessible semantic attributes, e.g., titles, authors, sentiments and tags. To address this issue, we build three text datasets with a diversity of semantic attributes, and propose Semantic Attribute Modulated (SAM) language modeling, a novel language modeling framework that incorporates various attributes. Attributes are selected automatically with an attribute attention mechanism. We empirically examine the language model perplexities of several typical corpora, and demonstrate the superiority of our model with different combinations of the attributes. Lyric generation by taking both the title and author attributes into account further demonstrates the effectiveness of our method.

1 Introduction

Language models (LMs) represent the probability distributions of the word sequences [Rosenfeld, 2000], which is a fundamental task in the NLP field. LMs have also been used in many NLP applications, such as speech recognition [Chan *et al.*, 2016], machine translation [Koehn, 2009; Cho *et al.*, 2014a; Bahdanau *et al.*, 2014], information retrieval [Manning *et al.*, 2008], etc.

Count-based language modeling is challenging since most of the possible word sequences are not seen before in practical applications. To alleviate this sparsity problem, it is natural to assume the word probability only depends on the previous $n-1$ words, which is known as the *n*-gram language model. Later on, Bengio et al. [2003] developed a feed-forward neural network language model and Mikolov et al. [Mikolov *et al.*, 2010] used the recurrent neural network (RNN) to train a language model. With the large-scale data and the modified gating function, RNN is now the most widely used model for the language modeling task [Chung *et al.*, 2014; Jozefowicz *et al.*, 2016; Dauphin *et al.*, 2016].

The recurrent neural network language models (RNNLMs) have demonstrated a good capability in modeling word probabilities, but are often criticized for incapable of capturing the long-term dependency, resulting in losing contextual information. Although the performance is in debate [Daniluk *et al.*, 2017], it is shown that the RNNLMs can be enhanced with some specific long-term contextual information, including LDA document topics [Mikolov and Zweig, 2012; Ghosh *et al.*, 2016; Dieng *et al.*, 2017], bag-of-words contexts [Wang and Cho, 2016], a neural memory cache [Grave *et al.*, 2017] and RNN sentence embeddings [Ji *et al.*, 2015]. Special RNN connections were also developed for different document structures to obtain useful information, such as the hierarchical RNNLM [Lin *et al.*, 2015] for sequences of sentences, the RNNLM for tree-structured texts [Tran *et al.*, 2016] and the dialog context RNNLM [Liu and Lane, 2017; Mei *et al.*, 2017] for dialogs.

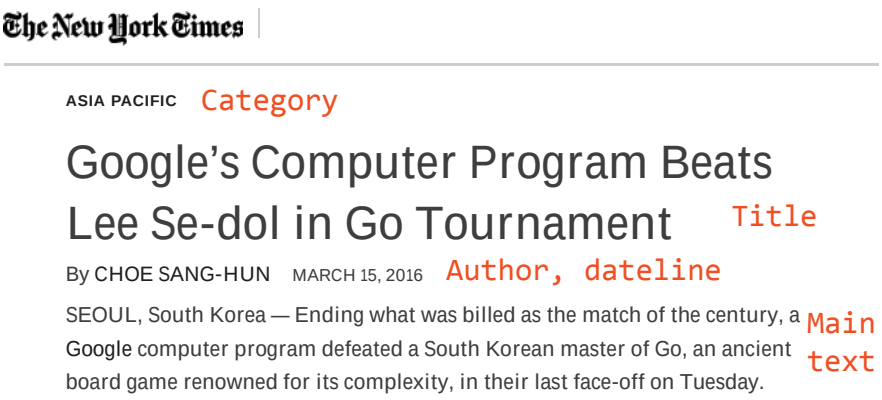

Figure 1: An AlphaGo News from NY Times have several important semantic attributes, such as the title, the author, the category and the dateline.

In the aforementioned models, only main text words were modeled and the vastly-accessible attributes of a document were omitted. However, in daily reading or speaking, some attributes implicitly convey key information of the word distributions and are often accessible before seeing the main texts. Document titles are compact abstracts carefully chosen by the authors. Labels and tags are specific assigned categories. Authorships reflect the writing styles and the attitude views.

In Figure. 1, we take an AlphaGo news article of New York Times[1] as an example. Given the title 'Googles Computer Program Beats Lee Se-dol in Go Tournament', the main text words 'Google', 'program' and 'Go' could be predicted more easily. Given the author attribute 'CHOE SANG-HUN' who is a Pulitzer Prize-winning South Korean journalist, we can better predict the words 'South-Korean' and 'Go'.

A more recent work gave a first trial to incorporate useful side information into the RNN-based language model [Hoang *et al.*, 2016], which adopted a bag-of-words representation of titles and key words. This bag-of-words representation does not have a strong representation capability and cannot utilize category-based attributes, such as authorships and document class labels.

1.1 Our Proposal

In this work, we present SAM, a Semantic Attribute Modulated language model. We incorporate vastly-accessible semantic attributes for language modeling and extract the attribute embedding using an attention mechanism. We construct three text datasets with attribute labels, covering formal news articles, forum comments and pop song lyrics. Our results show that with the attribute labels, our model better captures the semantic contents of documents, resulting in lower per-word perplexities. To demonstrate the effectiveness of our model, we show some interesting lyric generations with different title and author attributes.

In summary, our contributions are as follows:

- We build three text datasets, including formal news articles, forum comments and casual pop song lyrics, with the title and author attributes.
- We present SAM, a Semantic Attribute Modulated language modeling framework, in which a diversity of semantic attributes can be incorporated via an attention mechanism.
- By incorporating the selected semantic attributes, our model get better word prediction results on two typical corpora (PTB and BBCNews) and three collected corpora (IMDB, TTNews and XLyrics).

[1] https://www.nytimes.com/2016/03/16/world/asia/korea-alphago-vs-lee-sedol-go.html

- Based on our model, we generate some pop song lyrics with different title and author attributes, which further demonstrate the effectiveness of our method.

2 Preliminaries

In this section, we list the previous related works, including the recurrent neural network language models and their contextual extensions.

2.1 RNN-LM

Given a sequence of words $x = (x_1, x_2, \cdots, x_n)$, language modeling aims at computing its probability $P(x)$ by

$$P(\mathbf{x}) = P(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} P(x_i|x_{<i}), \tag{1}$$

where $x_{<i}$ are the words ahead of x_i. We can use the recurrent neural network to build a language model [Mikolov *et al.*, 2010]. At each time step i, the transition gating function ϕ reads one word x_i and updates the hidden state $\mathbf{h}_i$ as

$$\mathbf{h}_i = \phi(w_i, \mathbf{h}_{i-1}), \tag{2}$$

where $w_i = E^\top x_i$ is the continuous vector representation of the ont-hot input vector x_i and E is the embedding matrix. Commonly-used recurrent gating functions include the long-short term memory (LSTM) [Hochreiter and Schmidhuber, 1997] and the gated recurrent unit (GRU) [Cho *et al.*, 2014b]. The probability of the next possible word x^* in the vocabulary V is computed by

$$p(\hat{x}_{i+1} = x^*) \propto \exp((\mathbf{W}_h \mathbf{h}_i + b)_{x^*}), \tag{3}$$

where $\mathbf{W}_h \in \mathbb{R}^{|V| \times d}$, $b \in \mathbb{R}^{|V|}$ are the affine weights and biases respectively and d is the dimension of the word hidden state $\mathbf{h}_i$.

2.2 RNN-LM with Language Contexts

The RNN models are often criticized that they are not capable of capturing the long-term sequence dependence, resulting in unsatisfactory performance on modeling contextual information. Several previous works tried to capture the contextual information using the previous contexts. Let $f(x_{<i})$ be the contextual representation extracted from the contexts and the generation process of the RNNLM with $f(x_{<i})$ is

$$P(\mathbf{x}) = P(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} P(x_i|x_{<i}, f(x_{<i})).$$

The previous context representation $f(x_{<i})$ can be extracted as the bag-of-words contexts [Wang and Cho, 2016], the latent topics [Mikolov and Zweig, 2012; Ghosh *et al.*, 2016; Dieng *et al.*, 2017], the recurrent neural network embeddings [Ji *et al.*, 2015] and a neural memory cache [Grave *et al.*, 2017].

3 Semantic Attribute Modulated Language Model

Other than main texts, documents have semantic attributes, such as titles, authorships, tags and sentiments. These attributes provide important semantic information for the whole document. The semantic attribute modulated embedding represents the main document theme extracted from the title attribute, the writing style

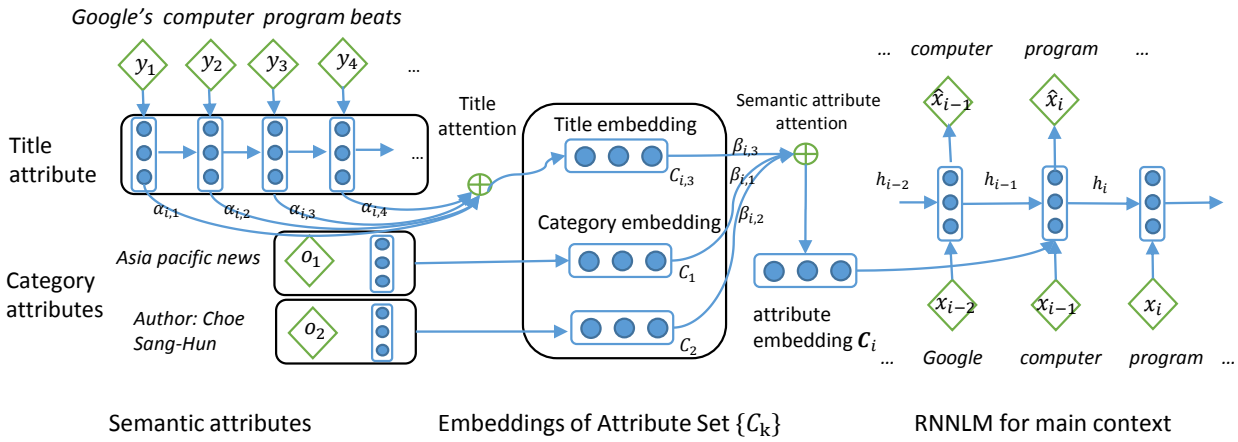

Figure 2: The SAM architecture

extracted from the authorship attribute and the semantic information from the categorical label attribute. In this section, we introduce SAM, the Semantic Attribute Modulated language model. First, we obtain the attribute set of the documents and get the attribute embedding via the semantic attention mechanism. Then, we develop the framework of the language model with the semantic attribute module.

Given the semantic attribute modulated representation $\mathbf{C}$, the generative process of our model is $P(x_i|x_{0:i-1}, \mathbf{C}), i = 1, 2, \cdots, n$, where x_i are the words in the same document.

As can be seen in Figure. 2, we build a Semantic attribute Modulated language model by incorporating the semantic attributes and the corresponding attention mechanism.

3.1 Semantic Attributes

Due to the discrepancy in the form of semantic attributes, we use different methods to extract representations from them.

Title Attribute The title is often carefully chosen by the author and is a compact summary of the document. In the language modeling framework, the titles have a different word distribution with the main texts but imply the semantic meaning for the main texts. Given an m-length title sequence $\mathbf{y} = (y_1, y_2, \cdots, y_t, \cdots, y_m)$, we use a recurrent neural network to extract the hidden state ϑ_t of every title word y_t as

$$\vartheta_t = \phi(E^\top y_t, \vartheta_{t-1}), \tag{4}$$

where the dimension of the title word hidden state is $\tilde{d}$.

Since title words do not have equal contribution to the whole context embedding, we use an attention mechanism for the title attribute, and obtain the different title representation c_i for different main text words x_i as a weighted sum:

$$C_i = \sum_{t=1}^{m} \alpha_{t,i} \vartheta_t,$$

where $\alpha_{t,i}$ is the attention value of the title word y_t for the main text word x_i,

$$\alpha_{t,i} = \frac{\exp(a(\vartheta_t, \mathbf{h}_{i-1}))}{\sum_{t=1}^{m} \exp(a(\vartheta_t, \mathbf{h}_{i-1}))}, \tag{5}$$

$\mathbf{h}_{i-1}$ is the hidden state of the previous time step in the main text, $a(\vartheta_t, \mathbf{h}_{i-1})$ is an attention function which scores how the title word y_t affect the main text word x_i. With this title attention, we automatically learn different importance weights of all title words for different main text words.

Category Attributes Category attributes are commonly used in daily writing and speaking. Useful category attributes includes document categories, authorships, sentiments, etc. We formulate the category attribute as a one hot vector o_k and the embedding of the category attribute is counted via an encoder of the one-hot vector $C_k = o_k \cdot e_k$, where e_k is a weight matrix which maps the one-hot vector to a continuous category embedding. We use the same embedding dimension for category attributes with the dimension of the title embedding as $\tilde{d}$.

3.2 Semantic Attribute Modulated Language Modeling

By using the above semantic embedding extractions, we obtain a set of semantic attribute embeddings $\{C_k\}$. To leverage the importance of each attribute for a main content word x_i, we adopt a semantic attribute attention mechanism to learn the semantic attribute embedding $\mathbf{C}_i$ for different main text words x_i as

$$\mathbf{C_i} = \sum_k \beta_{k,i} C_k, \tag{6}$$

$$\beta_{k,i} = \frac{\exp(b(C_k, \mathbf{h}_{i-1}))}{\sum_k \exp(b(C_k, \mathbf{h}_{i-1}))}, \tag{7}$$

where $b(C_k, \mathbf{h}_{i-1})$ is an attention function which scores how the attribute k affect the main text word x_i.

We incorporate the obtained semantic attributes into the RNNLM framework [Mikolov *et al.*, 2010] defined in Sec. 2.1. By using an attribute attention mechanism, at each time, the transition of RNNLM hidden state $\mathbf{h}_i$ not only reads the current word but also the semantic attribute embedding $\mathbf{C}_i$. Specifically,we concatenate the semantic attribute embedding $\mathbf{C}_i$ and the input word embedding vector $E^\top x$. Thus, the hidden states update as:

$$\mathbf{h}_i = \phi(w_i, \mathbf{h}_{i-1}), w_i = [E^\top x_i, \mathbf{C}_i]. \tag{8}$$

For the recurrent neural network funtion ϕ, we use the gated recurrent unit (GRU) [Chung *et al.*, 2014], since it has fewer gating connections. The GRU cell has two gates and a single memory cell and they are updated as:

$$\begin{aligned}
\text{update gate: } z_i &= \sigma(\mathbf{W}_z w_i + \mathbf{U}_z \mathbf{h}_{i-1}) \\
\text{reset gate: } r_i &= \sigma(\mathbf{W}_r w_i + \mathbf{U}_r \mathbf{h}_{i-1}) \\
\text{cell value: } \tilde{h}_i &= \tanh(\mathbf{W}_h w_i + \mathbf{U}_h(\mathbf{h}_{i-1} \odot r_i)) \\
\text{hidden value: } \mathbf{h}_i &= (1 - z_i)\tilde{h}_i + z_i \mathbf{h}_{i-1}
\end{aligned}$$

where σ is the sigmoid function and $\odot$ is the Hadamard product. Our model is trained by maximizing the log-likelihood of the corpus, using the back-propagation through time (BPTT) method [Boden, 2002].

Table 1: Statistics and Parameters of PTB, BBCNews, IMDB, TTNews, XLyrics

	PTB	BBCNews	IMDB	TTNews	XLyrics
#training docs	-	1,780	75k	70k	3.6k
#training tokens	923k	890k	21m	30m	118k
#vocabulary	10k	10k	30k	40k	3k
attribute(s)	Category	Title+Category	Title	Author+title	Author+title
hidden size	200	1000	1000	1000	1000

4 Discussions and Related Work

Contextual RNNLM Our work is related to several contextual language modeling works. Especailly Hoang *et al.* proposed to represent the titles and keywords as a bag-of-words representation and inject it to a general RNNLM model There are several major advantages of our paper over their method. First, we adopt a more diverse attribute set, including the widely used category attributes. Second, we use better attribute representation method, including a semantic attention mechanism and a RNN encoder. Third, we do the lyric generation task as a demonstration of how semantic attributes affect the language generations.

Note that the bag-of-words representation should work when the attribute have long texts. As can be seen in [Hoang *et al.*, 2016], using the bag-of-words represention, adding the long descriptions has better improvement than adding the short titles. We claim that the bag-of-words attributes can be easily incorporated in our SAM model and the insight of our paper should be: using different representation methods for different kinds of attributes.

Neural Machine Translation Neural machine translation (NMT) use the encoder-decoder network to generate specific response [Cho *et al.*, 2014a]. In NMT, the encoder network reads some source texts of one language, encode them into continuous embeddings and then decode network translate them into another language. NMT is also used to generate some poems after encoding some keywords [Wang *et al.*, 2016]. This is similar to our work as generate some texts given some useful attributes. The difference is that our work uses a semantic attribute attention to extract the semantic embedding instead of an encoder-decoder framework.

External Knowledge for Language Modeling Some useful side information is external, such as the knowledge base and some text references [Yang *et al.*, 2016; Ahn *et al.*, 2016]. Compared with these methods, our model concentrates on the document itself and does not depend on the external knowledge.

5 Experiments

In this section, we first give the settings of the datasets and experiments, and then we show the comparisons of the word prediction results between our method and several baseline methods. Using our model, we demonstrated some generated lyrics, with various title and author attributes.

5.1 Datasets

We evaluate the proposed language model with semantic attribute attention on five different dataset with different attribute combinations. Among these datasets, TTNews corpus, XLyrics corpus and title information for IMDB are collected by the authors. For detailed statistics, see Table. 1.

Penn TreeBank (PTB) Penn TreeBank (PTB) is a commonly-used dataset for evaluating language models and its texts are derived from the Wall Street Journal (WSJ). We use the preprocessed corpus by Mikolov *et al.* and it has 929k training tokens with a vocabulary of size 10k [2]. We use LDA topic model to analyze the PTB corpus with the topic number as 5. We assign the largest topic weight of each document as this document's category. The analysis of this category attribute and more discussions can be seen in Appendix A.

BBCNews BBCNews is a formal English news dataset and contains 2,225 BBC news articles collected by Griffiths *et al.*[3]. The BBCNews documents have 5 class labels: business, enternament, politics, sport and tech.

[2] http://www.fit.vutbr.cz/~imikolov/rnnlm/simple-examples.tgz

[3] http://mlg.ucd.ie/datasets/bbc.html

IMDB Movie Reviews (IMDB) IMDB Movie Reviews (IMDB) is a movie review corpus prepared by Maas *et al.* and has 75k training reviews and 25k testing reviews[4]. Note that Maas *et al.* did not provide review titles and we collected the titles according to the provided web links.

TTNews TTNews is a Chinese news dataset crawled from several major Chinese media[5]. TTNews has 70,000 news articles with 30 million tokens and a vocabulary of size 40k. Each document contains each contains title and author annotations.

XLyrics XLyrics is a Chinese pop music lyric dataset crawled from the web. XLyrics has 4k lyrics, about 118k tokens and a vocabulary of size 3k.

5.2 Experimental Setting

We consider several variants of the proposed methods with different combinations of semantic attributes. In detail, we consider the language modeling with a) a category attribute, b) a title attribute and c) a title attribute plus a category attribute.

We train a recurrent recurrent language model without any side information as a baseline method. We also report the results of a count-based n-gram model with the Kneser-Ney smooth method [Heafield, 2011].

For each recurrent neural network, we use the gated recurrent unit (GRU) as its recurrent neural network function [Cho *et al.*, 2014b]. Word Embedding dimension is set to be the same as the hidden size of RNN. Detailed parameter setting for each dataset is listed in Table. 1 For training, we use the ADAM method with the initial learning rate of 0.001 [Kingma and Ba, 2014] to maximize the log-likelihood and use early-stop training based on the validation log-likelihood.

5.3 Word prediction

Table 2: Word predictions that is improved, alike and worse after adding category attribute in Category Politics

	Words in the documents of the politics category
Improved	to, be, **ireland**, **bush**, one, **chairman**, **fiscal**, week, in, or, **plan**
Alike	both, general, many, both, but, is, N, in, been, the, said
Worse	of, gm, stock, orders, **law**, jerry

5.3.1 Language Modeling with Category-Attribute

Document categories implicitly points out the field of the word distribution. Document categories are indicative of the topic of discourse and therefore of the distribution over word. We first consider doing language modeling with category attribute on two corpora, PTB and BBCNews. For the PTB dataset, we use the LDA topic model to analyze the semantic information and we set the category as the largest topic weight for every document. The analysis and details can be seen in Appendix A. For the BBCNews dataset, we use the document class labels provided as a discrete category attribute.

In Table. 3, 5-Gram represents the count-based 5-gram model, RNN represents the conventional RNN model without any semantic attribute and SAM-Cat is our SAM model with a category attribute. As can be seen in the results, by adding a semantic category attribute, SAM-Cat outperforms the baseline models by achiveing lower perplexities.

[4] `http://ai.stanford.edu/œamaas/data/sentiment/`

[5] `http://www.ywnews.cn/`, `http://www.toutiao.com`, `http://www.huanqiu.com/`, etc

In order to discover why SAM-Cat outperforms traditional methods, we demonstrated the words in the politics category with the largest and the least perplexity change in Table 2, where we mark the words that have a strong semantic information of the politics in bold. The word predictions that is the most improved are generally related to the politics, the word predictions that have the least change are almost function words and most of the words prediction which have the largest degradation have a semantic meaning which is not related to the politics. We put the results of other categories in Appendix B. This result shows that by adding a category attribute, our SAM model better captures the semantic category information.

Table 3: Corpus-level perplexity with Category-Attribute on (a) Penn Tree Bank and (b) BBCNews

	PTB	BBCNews
5-Gram	141.2	131.1
RNN	117.1	76.7
SAM-Cat	**113.5**	**73.8**

5.3.2 Language Modeling with Title-Attribute Attention

Document titles are carefully chosen by the authors to summarize the document content and attract reader's attention. In this part, we incorporate the title attribute to take advantage of the implicit word distribution represented by the title. We use two corpora for this task, BBCNews as a formal published corpus and IMDB as a casual movie review corpus.

We implement the 5-gram model and the conventional RNN model on the corpus texts without titles. RNN-State is the conventional RNNLM model with the title's last hidden state as initialization. This means the title is considered as the first sentence, but isn't included into the evaluation of per-word perplexity. SAM-Title refers to the RNNLM model with an incorporated title attribute but no attention mechanism, i.e., each title words have an equal contribution to the title embedding. The SAM-Title-Att method is the SAM model with the title attribute and the attention mechanism. By adding the title's last hidden state to SAM-Title-Att as initialization, we get the SAM-Title-Att method.

We show the word prediction perplexity results in Table. 4. For the RNN-based models, adding the title embedding has better perplexity results. Moreover, SAM-Title is better than RNN-state because the injected title information would disappear after the non-linear gating functions. The attention-based title attribute performs better than the one without attention. This is because the attention mechanism provides the different importance weights for the title words. To further investigate how attention values control the importance weights, we visualize some of the attention values in Figure. 3. The color depth shows the attention weights. The red rectangles shows the title word 'Microsoft' has a large affect on the content words 'software' and 'unauthorised'. The title word 'move' has a large affect on the content word 'prove'.

Generally, our SAM model with title attribute performs better on BBCNews, compared with IMDB. We believe the result is caused by the different genres of these datasets. In order to make our title attribute useful, titles should be able to convey refined summaries of documents. BBCNews, as a formal news corpus written by professional journalists, usually has titles with higher quality than IMDB corpus.

5.3.3 Language Modeling with Title-Author-Attribute-Attention

In this part, we incorporate two different attributes, title and author. We will demonstrate that these two attributes are complementary and have different semantic information.

We use the semantic attribute attention to conjoin the two attributes and the suffix 'Au' means that this method incorporate the author categorical attribute and other method notations are the same with Table. 4.

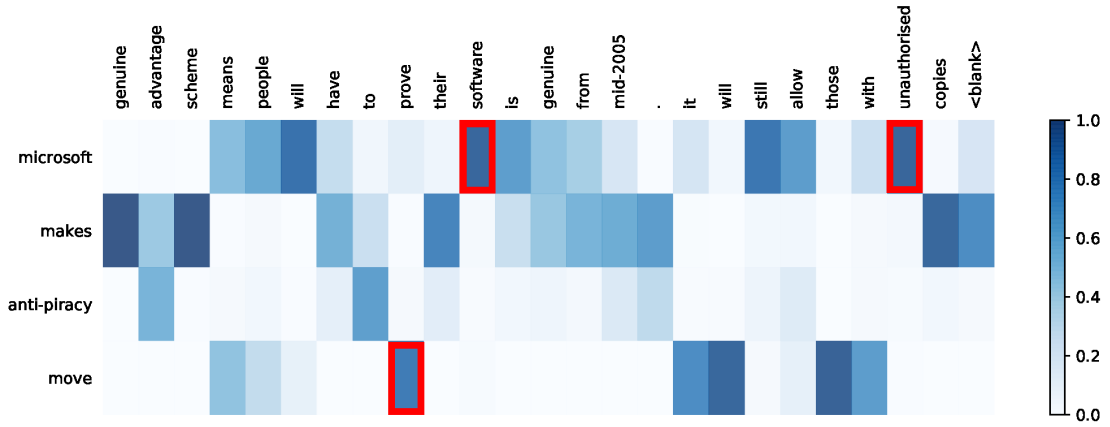

Figure 3: An example of alignment matrix from SAM-title

Table 4: Corpus-level perplexity with Title-Attribute on (a) BBCNews and (b) IMDB

	BBCNews	IMDB
5-Gram	131.1	124.6
RNN	76.7	62.6
RNN-State	72.2	61.0
RNN-BOW	72.2	61.8
SAM-Title	73.3	61.8
SAM-Title-Att	**71.3**	61.3
SAM-Title-Att-State	72.5	**60.9**

We show the word prediction perplexity results of several attribute combinations in Table. 5. For the TTNews and XLyrics datasets, we can see that incorporating both title and author attributes are better than a single one. The attention mechanism also help our method to choose the importance weights and the attention-based methods perform better than those without the semantic attention.

5.4 Language Generation with Semantic Attribute Attention

Many downstream applications of the language modeling task can be enhanced with the proposed semantic attributes. For machine translation, the semantic attributes could also be titles, authors and categories. For the speech recognition task, the semantic attributes include the age and the dialect of the speaker. For language generation task, such as the question-answering and the poem/lyric generation, the possible attributes are titles, authors and even styles.

We use the SAM model to perform lyric generation based on the XLyrics dataset and use both the title and author attributes. Given an original lyric, we generate a new one with the same title but a fake author. One example is demonstrated in Figure. 4. The original lyric *Your Face* is a sentimental love song written by *Xiaotian Wang* which is recalling the past love. After changing the authorship to *Lovely Sisters*, a trending Chinese band, we generate a joyful love song about the happiness of falling in love. Our generated lyrics have a similar style with some of the prized songs by *Lovely Sisters*. One example is *I say tonight's moonlight is beautiful, you say yes*[6], in which one lyric sequence is 'When we kiss each other/ I know this gonna not

[6] http://www.xiami.com/song/1771862045?spm=a1z1s.6639577.471966477.105.3HI96A

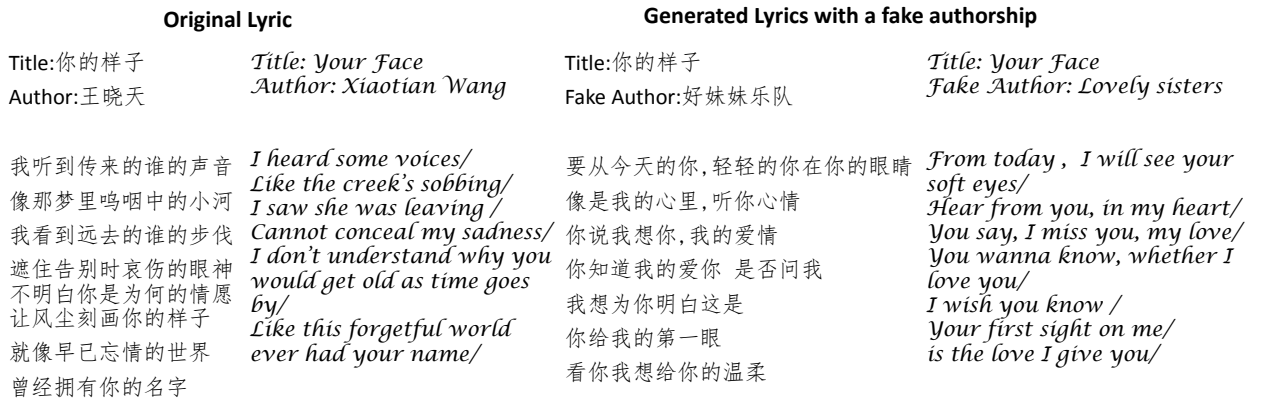

Figure 4: Generated lyrics with the same title but a fake authorship. The original lyric is sentimental (left) and the generated lyric with a fake author is cheerful (right).

Table 5: Corpus-level perplexity with Title-Author-Attribute on (a) TTNews and (b) XLyrics

	TTNews	XLyrics
5-Gram	136.7	8.13
RNN	120.1	7.56
RNN-BOW	118.4	8.18
SAM-Title-Att-State	118.1	7.30
SAM-Au-Att	117.2	7.24
SAM-State-Au	115.6	7.35
SAM-Title-Au	115.3	7.30
SAM-Title-Au-Att	114.1	7.08
SAM-Title-State-Au-Att	**113.4**	**6.84**

be right/ under the moon, this night/ we should take a risk and not lose each other'.

6 Conclusion

In this paper, we propose SAM, the semantic attribute modulated language model. The main idea is to take advantage of vastly-accessible and meaningful attributes to perform language modeling better. Our model adopts a diversity of semantic attributes including titles, authors and categories. With the attention mechanism, our model automatically selects the weights of the attributes. Experimental results demonstrate that our SAM model achieves better per-word predictions with selected semantic attributes. The visualization and the lyric generation results suggest that our model improved the performance by utilizing provided attributes to capture document's theme and writing style more easily.

In the future, we are interested in exploring more attributes which have semantic meaning when doing the language modeling task. In addition to the lyric generation task, other language generation tasks can also use our SAM model to utilize more semantic attributes. One possible example is to incorporate the position attribute into the speech recognition task to model the dialects.

7 Appendix A: Datasets Preparation of PTB

PTB is a commonly-used corpus benchmark for the language modeling task. We use the LDA topic model to extract semantic category attributes. Actually, adding a pseudo-category seems to be subtle for the language modeling task to see the words in advance and then predict them. We argue that the pseodu-category makes sense in the language modeling task evaluation for the following two reasons. First, We only add one discrete assignment for each document and there's no straightforward word distribution information propagated. Second, in fact, the category assignments have strong semantic information and we can find real category assignments for other datasets. The semantic analysis is as follows.

For the PTB dataset, we set the topic number as 5 and set the largest topic weight assignment as each document's category assignment. As can be seen in Table. 6, the topic $\#0$ focuses on the corporate finance, the topic $\#2$ focuses on the politics, the topic $\#2$ focuses on the managers, the topic $\#3$ focuses on the stocking market and the topic $\#4$ focuses on the daily news.

Table 6: Top words of 5 topics extracted from the PTB dataset

Topic	Top words
0	million billion share year company cents stock sales income revenue bonds profit corp.
1	its mr. federal company u.s. new government state court plan officials bill house
2	market stock trading prices stocks investors new price big index friday rates markets traders
3	its company mr. inc. new co. corp. president chief executive says group chairman business vice
4	mr. says when people years new time president work first few think good want city know back

8 Appendix B: More word predictions of the SAM-Cat on PTB dataset

In this part, we show some more word generations of our SAM-Cat model on the PTB dataset. We show that after adding the category attribute, we get more semantic word prediction improvements. In the main body of our paper, we show the results of the politics category. In Appendix B, we show the results on the categories 'corporate finance', 'managers' and 'stock market' in Table. 7.

Table 7: Word predictions that is improved, alike and worse after adding category attribute in categories 'corporate finance', 'managers' and 'stock market'

	Words in the documents of the corporate finance category
Improved	**exchange**, **share**, group, third-quarter, **soared**, from, is, **profit**
Alike	N,of, days, had, than, month, **share**, were, yield
Worse	reported, analysis, all, yield, vehicles, **economics**, gm, currently
	Words in the documents of the managers category
Improved	**market**,about,results,**orders**, **trading**, dow, **portfolio**, **price**, **market**
Alike	N, likely, of, prepared, southeast, futures, see, group, the
Worse	bear, totaled, optimistic, **executive**, chief, **manufacturers**, about
	Words in the documents of the corporate stocking market
Improved	**co**, **operating**, an, **markets**, considered, **commercial**, **stake**
Alike	N, usa, the is, s', these, discussion, at the, chickenb
Worse	offering, million, **money**, read, communications, lines, issues, city

References

[Ahn *et al.*, 2016] S. Ahn, H. Choi, T. Pärnamaa, and Y. Bengio. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*, 2016.

[Bahdanau *et al.*, 2014] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Bengio *et al.*, 2003] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[Boden, 2002] M. Boden. A guide to recurrent neural networks and backpropagation. *the Dallas project*, 2002.

[Chan *et al.*, 2016] W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE, 2016.

[Cho *et al.*, 2014a] K. Cho, B. Van M., C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Cho *et al.*, 2014b] K. Cho, B. Van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[Chung *et al.*, 2014] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Daniluk *et al.*, 2017] M. Daniluk, T. Rocktäschel, J. Welbl, and S. Riedel. Frustratingly short attention spans in neural language modeling. In *ICLR*, 2017.

[Dauphin *et al.*, 2016] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

[Dieng *et al.*, 2017] A. B. Dieng, C. Wang, J. Gao, and J. Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. In *ICLR*, 2017.

[Ghosh *et al.*, 2016] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck. Contextual lstm (CLSTM) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*, 2016.

[Grave *et al.*, 2017] E. Grave, A. Joulin, and N. Usunier. Improving neural language models with a continuous cache. In *ICLR*, 2017.

[Griffiths *et al.*, 2004] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B Tenenbaum. Integrating topics and syntax. In *NIPS*, volume 4, pages 537–544, 2004.

[Heafield, 2011] K. Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.

[Hoang *et al.*, 2016] C. Hoang, G. Haffari, and T. Cohn. Incorporating side information into recurrent neural network language models. In *Proceedings of NAACL-HLT*, pages 1250–1255, 2016.

[Hochreiter and Schmidhuber, 1997] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Ji *et al.*, 2015] Y Ji, T. Cohn, C. Kong, L.and Dyer, and J. Eisenstein. Document context language models. *arXiv preprint arXiv:1511.03962*, 2015.

[Jozefowicz *et al.*, 2016] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

[Kingma and Ba, 2014] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Koehn, 2009] P. Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[Lin *et al.*, 2015] R. Lin, S. Liu, M. Yang, M. Li, M. Zhou, and S. Li. Hierarchical recurrent neural network for document modeling. In *EMNLP*, pages 899–907, 2015.

[Liu and Lane, 2017] B. Liu and I. Lane. Dialog context language modeling with recurrent neural networks. In *ICASSP*, 2017.

[Maas *et al.*, 2011] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[Manning *et al.*, 2008] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*. Cambridge university press Cambridge, 2008.

[Mei *et al.*, 2017] H. Mei, M. Bansal, and M. Walter. Coherent dialogue with attention-based language models. In *AAAI*, 2017.

[Mikolov and Zweig, 2012] T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. In *SLT*, pages 234–239, 2012.

[Mikolov *et al.*, 2010] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[Mikolov *et al.*, 2011] T. Mikolov, S. Kombrink, J. Burget, L.and Černockỳ, and S. Khudanpur. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE, 2011.

[Rosenfeld, 2000] Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

[Tran *et al.*, 2016] Q. H. Tran, I. Zukerman, and G. Haffari. Inter-document contextual language model. In *Proceedings of NAACL-HLT*, pages 762–766, 2016.

[Wang and Cho, 2016] T. Wang and K. Cho. Larger-context language modelling with recurrent neural network. In *ACL*, 2016.

[Wang *et al.*, 2016] Z. Wang, W. He, H. Wu, H. Wu, W. Li, Wang H., and Chen E. Chinese poetry generation with planning based neural network. In *COLING*, 2016.

[Yang *et al.*, 2016] Z. Yang, P. Blunsom, C. Dyer, and L. Wang. Reference-aware language models. *arXiv preprint arXiv:1611.01628*, 2016.