Detecting Horrible Episodes: Hint Fiction as a Case Study

Yong Ren  
Graduate School of Information Science and Technology, The University of Tokyo  
renyong@tkl.iis.u-tokyo.ac.jp

Naoki Yoshinaga  
Institute of Industrial Science, The University of Tokyo  
yoshinaga@tkl.iis.u-tokyo.ac.jp

Nobuhiro Kaji  
(affiliation as previous author)  
kaji@tkl.iis.u-tokyo.ac.jp

Masaru Kitsuregawa  
National Institute of Informatics; Institute of Industrial Science, The University of Tokyo  
kitsuregawa@tkl.iis.u-tokyo.ac.jp

keywords: horror identification, unsupervised learning, unlabeled corpus

Summary

Nowadays, the task of predicting readers’ perception for given text has attracted much interest in natural language processing area. In this study we explore identification of horror episodes, which is helpful to avoid triggering bad experience to users. We formulate horror recognition as a binary classification problem. Besides bag-of-words (BoW), we attempt to exploit features derived from a large scale of unlabeled corpus in this task. We investigate the usefulness of the beyond-BoW features through a series of experiments, using newly emerging literature, hint fiction, as typical, concise example of horror episodes.

1. Introduction

Recently, reader emotion prediction has attracted much attention from NLP researchers [Lin 07, Lin 08, Yano 09, Hasegawa 13]. Usually, the readers’ emotion can be classified into several universal types including anger, disgust, fear, happiness, sadness, and surprise [Ekman 75]. It is generally believed that such kind of study can be helpful to better learn the customers; for example, [Wang 13] has confirmed the readers’ reaction triggered by words will play important role in the click-through rate of online advertisement.

Though there are several pioneers on classifying readers’ emotion, the task of discerning one specific reaction from the others has barely been studied. Here, we focus on detecting episodes that can trigger horrible emotion. We found that such episodes are spreading widely in the SNS, and causing bad user experience. For example, the following text (originally in Chinese, but manually translated into English) is a representative horror story in Chinese microblog platform. It was once retweeted more than one thousands times.

Since people who living in high-rise apartment usually use elevators, the stairs become the unnoticed place. One night, a girl who lived in 13th floor wanted to came back home. Unfortunately, the elevator was not in service due to its malfunction. Seeing the long stairs, she was scared. So the girl let her mother come downstairs to pick her up. Her mother came, and the girl went upstairs with her mother. At 12th floor, the girl received the phone call from her mother,’I have just arrived. Where are you?’.

After a brief thinking, readers could realize that the girl in this story was possibly taken away by a ghost other than her mother. We believe that similar stories will arouse bad image from many readers, especially, those people who have children. Identifying those episodes in advance can prohibit them touching those improper readers.

In our study, we formulate horror identification as a binary classification problem, and employ widely-adopted supervised learning algorithm, Supported Vector Machine (SVM) [Vapnik 95], to resolve it. Besides bag-of-words (BoW), we also investigated the impact of features derived from a large scale of corpus\(^1\). Through the empirical evaluation, we confirm that those auxiliary features from

\(^1\) In this study, we only refer the textual features in the original text as “bag-of-words” features.
the document collection can improve the performance. Finally, we exploit conventional feature selection to locate the significant features and analyze their relation to horror.

We conduct experiments on hint fiction, which is a newly emerging literature in microblog platform. Since terrible hint fiction is the typical source that can evoke readers’ horrible emotion.

The rest of this paper is organized as follows. In Section 2, related work is introduced. In Section 3, we explain the feature learning algorithms we adopted. In Section 4, we evaluate the usefulness of induced features. In Section 5, we conclude this study and outline our future direction.

2. Related Work

[Lin 07] initiated the task of sentiment analysis on readers’ standpoints. They exploited Naive Bayes and SVM to categorize news articles according to readers’ emotion that is evoked by them, where the emotions are happy, angry, sad, surprised, heartwarming, awesome, bored and useful. They investigated the influence of different feature combinations among Chinese character bigrams, words, metadata of articles and the emotion dictionary, and concluded that combination of all those features contributed to the best performance.

[Tang 11] explored how the writer emotion affects readers’ emotion. Their evaluation was conducted on a microblog dataset. Besides textual features, they also utilized three kinds of non-linguistic features: social relation between the writer and the reader, reader behavior, and relevance between the original post and the comment. They found that those beyond-text features are helpful in predicting the emotion of readers. Similarly, [Hasegawa 13] expanded the domain to addresser/addressee in online dialogue. They induced additional features from the lister’s previous utterance, which is uniquely available in the dialogue dataset.

Different from those previous study using supervised learning approaches, [Yano 09] extended latent dirichlet allocation (LDA) [Blei 03] to capture the blogosphere characteristics such as the authorship and reader reaction, and then they make prediction on the response for the political blog passages by using the proposed model. They discovered that the topic model is promising in predicting the reader emotion.

In terms of the study focusing on discerning one specific perception from the others, [Mihalcea 05] made use of Naive Bayes and SVM to distinguish humorous text from non-humorous text. They only take one sentence joke (“one-liner”) into account. The excellent result obtained in distinguishing humorous one-linear from new titles or from proverbs. However, when the negative instances are British National Corpus (BNC) which are text in mixed form, the performance highly degenerates, which indicates the difficulty in capturing humor in text.

3. Proposed Method

The shortage of context drive us to induce auxiliary features from a large scale of corpus. Actually, deriving features from unlabeled document collection has attracted increasing interest in the NLP field [Turian 10, Lin 09]. These techniques alleviate the data sparsity issue, and many works have demonstrated excellent results in corresponding tasks [Miller 04, Koo 08].

In this section, we first introduce the unsupervised learning methods we adopted, then make a comparison between them using instances.

3.1 Brown Clustering

Clustering words according to their context (surrounding words distribution) is an ordinary method to provide additional features for those words. One straight idea is including the cluster IDs of words as auxiliary features, which is also explored in our study. We take measures of the representative hierarchical words clustering algorithm, Brown clustering [Brown 92], in this task. Brown clustering was also exploited in many NLP tasks such as named entity recognition (NER) [Miller 04] and dependency parsing [Koo 08].

The Brown-clustering is based on class-based bigram language model as Figure 1 shows, where $w_i$ is one specific word and $c_i$ specifies the corresponding cluster. We can clearly find that Brown clustering supposes that the
cluster of one word is affected by the previous one. More precisely, given one clustering function \( C(C(w_i) = c_i) \), the quality metric of the function \( C \) is defined as follows [Liang 05]:

\[
\sum_{c, c'} P(c, c') \log \frac{P(c, c')}{P(c)P(c')} + \sum_w P(w) \log P(w) \quad (1)
\]

Formula (1) is the actual objective that Brown clustering tries to maximize. Here, \( c \) and \( c' \) are two consecutive clusters in the text. Interested readers can find the detailed derivation in [Liang 05]. Note that the time complexity of Brown clustering is \( O(NC^2) \), where \( N \) is the number of words and \( C \) is the number of clusters.

3.2 Neural Network Language Model

Different from the conventional study that assumes one word to correspond to one dimension in the (sparse) feature space, neural network language model (NNLM) [Bengio 06] represents each single word as a dense vector. Moreover, the representation is learnt automatically by an unsupervised method. No feature refinement or designing process exists during the learning phase. It is generally believed the word feature learnt could capture multiple degrees of similarity [Mikolov 13b]. We use word2vec [Mikolov 13a], the state-of-the-art implementation of NNLM, to obtain a vector representation for each word. Two models, continuous bag-of-words model (CBOW) and continuous skip-gram model, are proposed in the framework. Here, we adopted the latter since a better performance has been reported in [Mikolov 13a].

Figure 2 displays the sketch of continuous skip-gram model. We can clearly observe that the critical point is to predict the words within the window of given word \( w_t \). In our evaluation, we set the window size to five ( \( t = 2 \) ). A commonly adopted soft-max classifier is used during the word prediction. One attracting point is the capability to conduct the training on the corpus with billions of words. We will further exploit the usage of word2vec in our future work. To the best of our knowledge, word2vec was barely explored in NLP tasks.

4. Empirical Study

4.1 Setting

Currently, our target domain is hint fiction, which is a newly emerging form of literature in microblog platform. We collected hint fictions from the website\(^2\) which lists various genres of hint fictions. We only kept three types, horror, humor, and moved, since they are closely related to reader emotion. Table 1 report the statistics on the dataset. We find that the average number of words in each text is around 130, which conforms to length limit in microblog platform.

We randomly separated each type of those text into five equally-sized parts, and then combine them accordingly. Eventually, there are five subsets and each of them are nearly balanced in the classes. In this evaluation, we take those horror and moved text as non-horror instances. We employed five-fold crossing validation alike strategy. Note that in each round we only use one subset as training data, and treat the four subsets left as test data. The motivation of this strategy locates in the fact that labeling data is time consuming and labor intensive, and we cannot guarantee the number of training data available is surely more than the test counterpart in practice.

The unlabeled corpus is provided by datatang\(^3\). It consists of 14 million Chinese web text, and spans in time from 1992 to 2011. During the current evaluation, we utilized the 2011 samples, which contains 941,157 documents with more than 3 million words. We plan to put all the corpus into usage in our future work.

4.2 Feature Learning Comparison

In our study, we set the number of clusters to 256 in Brown clustering, and we use the optimized version\(^4\) as the implementation, but it still take almost thirty hours to finish the clustering on our experimental corpus (intro-

\(^2\) http://s.gdxsw.cn/

\(^3\) www.datatang.com

\(^4\) https://github.com/percyliang/brown-cluster
duced in Section 4.1). During the NNLM training phrase, we adopted the implementation provided by Google\(^5\). We adapted the Gensim version used\(^6\) when we carried out the reference, since it is convenient to customize the similarity computation based on the trained model. Note that we need to specify the dimension number for the word vector in the initialization of word2vec. Here, we set the value to 100.

Currently, we issued three words, “dead body”, “found” and “sudden”, into the results derived from Brown clustering and word2vec respectively. All the three words are among the top-10 statistically significant features (introduced in Section 4.4).

First, we observed that the number of words located in the same clusters with those three keywords are 52, 187 and 5, 748 and 3, 719 respectively. The related cluster did contain several closely related words. For example, “saw” and “hearted” are grouped with “found”. However, many verbs such as “walked away” appeared in this cluster, just because they perform syntactically similar. Worse, there is no metric to measure the similarity between two words.

In terms of neural language model, table Table 2 listed the top-10 related words derived from word2vec for the three input words. These words are ranked by the similarity score, for it is easy to compute the similarly between two vectors. We amazingly found that they are very semantically related with each other. For example, “carried away” and “bury” are verbs, but they show up in the “dead body” related list. Similarly, “screams” appear in the “sudden” related list.

In a short summary, while Brown clustering is good at capturing syntactically similar words, word2vec is a better choice to locate semantic related words. Since horror identification needs to understand semantics of the story and the hint fiction is brief, we directly appended top-10 similar words for each word in the text.

### 4.3 Performance Comparison

The purpose of our experiment is to investigate the usefulness of auxiliary features (cluster ID for Brown clustering, and top-10 similar words for word2vec) derived from the web document collection. We carried out the evaluation in the framework of binary classification, and used the widely-adopted supervised learning algorithm SVM [Vapnik 95] in this task.

Figure 3 the classification performance when changing the hyper-parameter C. We conclude that both the features learnt using Brown clustering and word2vec can promote the recall rate. We owe this to the additional related context provided by Brown clustering and word2vec. But the precision degenerates a bit due to the decrease on the discerning power of those features (analyzed in the following section). As a whole, there is moderate improvement on the F1 metric.

### 4.4 Significant Feature Exploration

In order to get the better understanding on those derived features, we investigate the significant ones. It is known that the features are not equal in distinguishing instances. Here, we use the conventional Chi-square feature selection [Singh 10] to locate the statistically significant features.

Table 3 lists the top-10 significant features and their relation to the horrible storylines. Even the feature selection we adopted is conventional and simple, we can see the amazing association between the significant features and the terrible scenes. Comparatively, Table shows the top-10 significant features derived from Brown clustering and word2vec respectively. Note that we appended the most frequent word in each cluster to each cluster ID in so that we can learn the word type in each cluster. In both cases, nouns are helpful to judge whether the text is ter-

---

\(^5\) https://code.google.com/p/word2vec/

\(^6\) http://radimrehurek.com/gensim/models/word2vec.html
rifying or not. One possible explanation is that the topics in terrible stories are different from the other genres. For example the authors of horrible hint fiction tend to establish the happenings to stairwell or night. In word2vec significant features, “found”-related words (“saw”, “met”, “heard” and “suddenly”) play critical role in discriminating horrible stories from the others. These two discoveries open the door for our future work. These words are used to describe things occur unexpectedly. We will pay much attention to the thematic information and particular verb collocation.

However, we should note that we cannot relay too much on those features derived. We can find that the relation between these auxiliary features and terrible plot are weaker than the original text features. For example, we can see the Proper Noun “Otis” (a brand name of elevator) also emerge as a highly significant features. For example the authors of horrible hint fiction tend to emphasize deriving on all the corpus in our hand and evaluate tential classes in the bunch of text where we need to identify those horrible stories. Our final goal is to realize the unsupervised horror detection. We believe we can learn a lot on word meaning such as topic belongings, semantic similarity from the large amount of corpus available.

5. Conclusion and Future Direction

In this study, we formulated the horror detection as a classification task, and made use of well-adopted supervised learning method to resolve it. Besides the bag-of-words features, we take measures of two unsupervised learning approaches, Brown clustering and word2vec to derive additional features from a large scale of unlabeled corpus. We conclude that we can benefit from these beyond-BOW features.

In our future study, we will first conduct the unsupervised deriving on all the corpus in our hand and evaluate our approach on a news dataset. The scalability and speed will be our new focus. Then, we will consider the one-class strategy, since it is not realistic to judge all the possible categories

**References**


[ 担当委員： ]

Received MM DD, 19YY.