# Table of Contents

**Abstract [Acknowledgements](#_30j0zll) [List of Tables](#_3znysh7)**

**[List of Figures](#_tyjcwt)**

1. **[Introduction](#_1t3h5sf)**

**II III VI VII**

**1**

1. **[Background and theory](#_2s8eyo1) 5**
   1. [Background](#_3rdcrjn) 5
   2. [Deep learning and NLP](#_lnxbz9) 6
   3. [Keyword extraction](#_44sinio) 7
      1. [Supervised keyword extraction algorithm](#_3j2qqm3) 8
      2. [Semi-supervised keyword extraction algorithm](#_2xcytpi) 9
      3. [Unsupervised keyword extraction algorithm](#_3whwml4) 10
   4. [Topic model](#_qsh70q) 11
      1. [LDA](#_1pxezwc) 12
      2. [LSI](#_147n2zr) 14
   5. [Text generation](#_23ckvvd) 14
      1. [General information](#_32hioqz) 14
      2. [Language model](#_vx1227) 17
      3. [Transformer](#_1v1yuxt) 17

[2.5.4 GPT-2](#_2u6wntf) 18

* 1. [Terms and concepts](#_3tbugp1) 18

1. **[Method, system design and development](#_nmf14n) 23**
   1. [Dataset and data preprocessing](#_1mrcu09) 23
   2. [Text vectorization](#_3l18frh) 25
   3. [System development](#_4k668n3) 27
2. **[Results and evaluation](#_3ygebqi) 30**
   1. [Results](#_sqyw64) 30
      1. [Keyword extraction](#_3cqmetx) 30
      2. [Story generation](#_2r0uhxc) 32
   2. [Evaluation method](#_25b2l0r) 32
      1. [Automated metrics](#_34g0dwd) 32
      2. [Human evaluation](#_43ky6rz) 34
3. **[Discussion and Conclusions](#_3hv69ve) 36**
   1. [Discussion](#_4h042r0) 36
   2. [Conclusion](#_1baon6m) 37

**[Bibliography](#_2afmg28) 38**

# List of Tables

[Table 3.1: Original text and contained information](#_46r0co2) 24

[Table 4.1: A sample output of Keyword extraction module](#_1rvwp1q) 31

[Table 4.2: A sample output of text generation module](#_1664s55) 32

[Table 4.3: Survey results](#_2iq8gzs) 35

# List of Figures

[Figure 2.1: Deep learning approach for NLP](#_35nkun2) 7

[Figure 2.2: General keyword extraction process](#_2jxsxqh) 8

[Figure 2.3: An example of supervised keyword extraction process](#_1y810tw) 9

[Figure 2.4: A three-layer Bayesian model](#_49x2ik5) 12

[Figure 2.5: Text generation with Recurrent neural network](#_1hmsyys) 15

[Figure 2.6: A example of bag-of-words model](#_41mghml) 16

[Figure 3.1: Example of separate stories](#_2lwamvv) 25

[Figure 3.2: The architecture of the storytelling system](#_2zbgiuw) 27

1. **Introduction**

With the rapid development of science and technology, the application of high technology tools has become increasingly popular in educational field. School budgets are shrinking nowadays, the number of students per class- room is increasing, and the demand for greater personalising of curricula is expanding, a need for technological support appears. Recent studies re- ported that intelligent social robots are being widely used in early childhood education[[1](#_pkwqa1)]. A common application of social robots is in education since they have been proved can effectively increase youngster’s cognitive ability and effective improvement on social communication[[2](#_39kk8xu)]. Social robots have been used as tutors and learning companions when personalized learning is needed, and also have made good performance as those human tutoring on limited range of tasks. Educational robotics is a representative of the application of robots in the field of education. Some studies[[3](#_1opuj5n)] have demon- strated that robot use’s positive impacts on children’s cognition, language, interaction, social and moral development. In present studies, researchers pointed out that educational robots usually play three roles, tutor, peer learner and supportive tools[[1](#_pkwqa1)]. Therefore, robots mainly play as a learning object, tool and companion in the teaching process. From the perspective of market development status, educational robot products are mainly used in domestic and school fields, for instance, acting as tutors or peer learners, children’s entertainment and education companions, and domestic intelli- gent assistants, special education robots for autism in training institutions. Educational robots will become a trend in the future. Nowadays, society needs talents with innovative consciousness and creative thinking, espe-

cially in the future.

In the past, robots replaced humans in completing industrial operations by inputting set-up instruction programs. With the development of sci- ence technology, Artificial Intelligence(AI)[[4](#_48pi1tg)] have been widely applied in robots. For instance, Automatic Speech Recognition(ASR)[[5](#_2nusc19)], emotion recognition[[6](#_1302m92)]. Robots nowadays are like being given an intelligent "brain" that is able to think, learn, and conduct tasks independently like a human being. At present, robots can be regarded as a representative application of artificial intelligence.

Artificial intelligence is playing a crucial role in the application of robots. It is impacting every industry and human being in many aspects. As an important field of artificial intelligence, intelligent social robots are entering schools in diverse degrees, playing an important role in intelligent assis- tance and multi-media classrooms.

Storytelling is a crucial part of the tutoring target, which can not only enhance children’s memory, but also affect positively on enlightenment education. Storytelling can effectively improve children’s expression skills, which is very important in a society where communication is becoming more and more intense. Secondly, storytelling has a positive impact on children’s emotional shaping progress[[3](#_1opuj5n)]. In addition, storytelling helps children expand their knowledge. The demand for educational technology support is increasing with the development of the population and economy. The reduction in school expenditures, the increase in the number of stu- dents, and the popularization of modern education have made education more and more demanding on the individualized curriculum, which has prompted the society to conduct research on technology-based support. Artificial intelligence has increasingly become a hot topic in society and education, which is of great significance to education. Researchers are also

exploring the combination of artificial intelligence and robots in the field of education. Recently, social robots have been proved to have a significant effect on improving cognitive reaction and have attained similar outcomes compared to that human education on daily tasks[[1](#_pkwqa1)].

However, a fully autonomous robotic system is particularly challenging, es- pecially when robots are introduced as conversational agents. In this sense, speech recognition and natural language understanding (NLU) is still diffi- cult to implement. Storytelling is one of the functions where robots serve as learning companions. Automatic storytelling is a challenging text gener- ation task since it requires generating long, coherent natural language to describe a sensible sequence of events. Many efforts have been contributed in automatic storytelling in the past, prior work either has limitations in plot planning, or simply use settled certain keywords to generate stories in a restricted domain[[7](#_3mzq4wv)].

In the past, Long Short Term Memory Network (LSTM)[[8](#_2250f4o)] is widely used in text generation, to address the problem that Recurrent Neural Network[[9](#_haapch)] losing the information of the input before several time points by introducing long-term and short-term memory channels controlled by gates. However, the limitations of LSTM includes that it is not suitable for transfer learning, and cannot be used for parallel computing, and the scope of attention is limited even after expansion.

The Transformer model[[10](#_319y80a)] directly discard recursive modeling. The differ- ence is that with the assist of the attention matrix, Transformer can directly access other elements of the output, so that they have an infinite attention interval. In addition, it can also perform parallel calculations.

In this project, we aim to develop an automatic storytelling system based on natural language processing technologies that extract keywords from input

sentences and then generate a new story with these keywords. The purpose of this project is to build an automatic system that can tell a story from given sentences, and the performance of the system need to be evaluated and the output should be evaluated how well they can be perceived by humans. The system consists of two main components: **keyword extraction module** and **story generation module**. The system includes a storytelling generator module that can automatically generate pieces of the story with a coherent natural language based on the sequence of events told by the child. Therefore, the problem is considered as a machine learning problem, specifically, a natural language processing problem, which requires a model to generate a story that is based on the given sentences.

Based on the functions requirements, we need to address the following problems: In the keyword extraction phase, **how to extract keywords from the given input, and how many keywords we need to extract**; in the text generation phase, **how to use the keywords to generate the new story** and in the evaluation phase, **how does the generation model perform in terms of perplexity and how are the stories perceived by human participants in terms of cohesion and coherence**.

This thesis is structured as follows: Section [2](#_2s8eyo1) covers the background and related work. Section [3](#_nmf14n) introduces methods used in the development, design of the system and the evaluation method in detail. In Section [4](#_3ygebqi), we describe and discuss the results and the evaluation results. In section [5](#_3hv69ve), we conclude results, limitations of the thesis and describe the future work.

# Background and theory

## Background

Natural language is a fundamental feature that distinguishes humans from other animals. Without language, expressing human thinking would be a difficult task, so natural language processing enables the machines to read, understand and derive meaning from human languages. Natural Language Processing (NLP)[[11](#_1gf8i83)] is a branch of AI. It is an interdisciplinary subject of AI and computer science and linguistics, also called computa- tional linguistics. It focuses on the communication between computers and humans in natural language. It describes the interaction between human language and computers. NLP makes computers to process, understand, and use human languages (such as Chinese, English, etc). With NLP, tasks such as automatic speech recognition and automatic text generation can be completed. And using computer resources to process large data (text) set tasks, such as automatic summarizing (generating a summary of a given text) and machine translation will be less time-consuming and cost less labor expenses.

NLP can be seen from two perspectives, research and application. From the perspective of the contents of research, nlp includes syntax analysis, semantic analysis, and text comprehension, etc. On the other hand, from an application point of view, natural language processing has broad appli- cation prospects[[11](#_1gf8i83)]. Especially in the era of information, applications of natural language processing are all-encompassing, such as machine transla-

tion, speech recognition, information retrieval, information extraction and filtering, text classification and clustering, public opinions analysis and opinions mining, etc, which involves data mining related to language pro- cessing, machine learning, knowledge acquisition, knowledge engineering, artificial intelligence research, and linguistic research related to language computing[[12](#_40ew0vw)][[13](#_2fk6b3p)].

Since NLP is the cross field of computer science, linguistics and machine learning. In short, NLP is dedicated to enabling computers to understand and generate human language text[[14](#_upglbi)]. NLP technology has been widely used in many fields, such as voice assistants such as Tmall Genie[[15](#_3ep43zb)] and Siri[[16](#_1tuee74)], as well as machine translation and text filtering. Machine learning is one of the most far-reaching areas affected by NLP, especially deep learning technology. The field can be divided into the following three parts:

* + 1. Speech recognition: translate spoken language into text.
    2. NLU: the ability of computers to understand humans.
    3. Natural language generation: computer-generated natural language.

## Deep learning and NLP

Deep learning is a branch of machine learning that tends to automatically learn appropriate features and their representations, and tries to learn multi- level representations and outputs[[17](#_4du1wux)]. It has considerable effects on the application of NLP, such as machine translation, sentiment analyze, ques- tion and answering systems, etc.

One of the important features of deep learning based NLP is to use vectors to represent elements at various levels. Traditional methods use sophisticated methods to label, while in deep learning based NLP, vectors are used to

represent words, phrases, logical expressions and sentences. The word vector combines the deep network and human text in a good way, and it is an important step for the neural network to "understand" the text. It uses a rule-based method to represent words as encoded vectors, for instance, one-hot representation and word-embedding. And then build a multi-layer neural network to learn independently[[18](#_2szc72q)]. NLP based on deep learning focus more on semantic representation than syntactic representation[[19](#_184mhaj)]. Figure[2.1](#_35nkun2) describes how general deep learning-based NLP works.

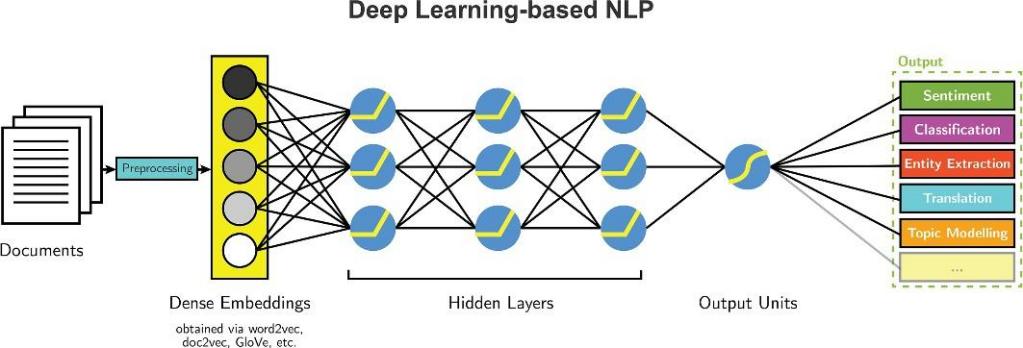


Figure 2.1: Deep learning approach for NLP

## Keyword extraction

In NLP, one of the most important thing to process massive text files is to extract the information that users are most concerned about. Regardless of whether it is a long text or a short text, the topic of the entire text can often be concentrated through a few keywords. Keywords are words that can express and convey the most central and crucial contents of a document. They are generally used in computer systems to index the features of papers, information retrieval, and system collection for readers to review.

Keyword extraction is a branch of the text mining field[[20](#_3s49zyc)]. It is the basic task of text mining research such as text retrieval, document comparison, abstract generation, document classification and clustering. At the same time, whether it is text-based recommendation system or text-based search engine, their dependence on text keywords is considerable. The accuracy of keyword extraction effects directly on the final outcomes of the recom- mendation system or the search engine such as Baidu[[21](#_279ka65)] and Bing[[22](#_meukdy)]. Consequently, keyword extraction is an important part in text mining. The figure [2.2](#_2jxsxqh)[[23](#_36ei31r)] presents a general process of keyword extraction[[24](#_1ljsd9k)]:

* + 1. Preprocessing the original dataset.
    2. Get the candidate keywords from the article.
    3. Get the topic model through large-scale corpus learning.
    4. Calculating the article topic model.
    5. Sorting. Calculate and sort the topic similarity between the document and the candidate keywords, and select top-k words as keywords.

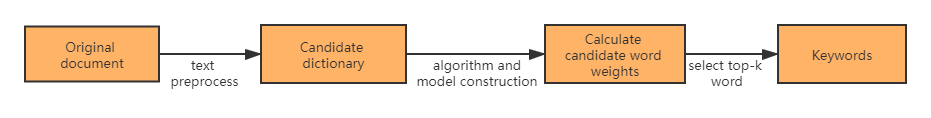


Figure 2.2: General keyword extraction process

According to some recent studies and researches[[25](#_45jfvxd)], there are three meth- ods for text keywords extraction.

### Supervised keyword extraction algorithm

This approach consider the keyword extraction process as a classification problem that to determine whether a word or phrase in a document is a

keyword or not. The keyword extraction process in this method is regarded as a binary classification problem. Since it is a classification problem, pro- viding the labeled training corpus to the model becomes imperative. The candidate words are extracted at first, and then mark the label for each candidate word, either marked as a keyword or not. After all the words are marked, a keyword extraction classifier will be trained. After adding a new document, all the candidate words will be extracted, and then use the trained keyword extraction classifier to classify each candidate word, and finally use the labeled candidate word as a keyword. Figure[2.3](#_1y810tw) describes an example of supervised keyword extraction process.



Figure 2.3: An example of supervised keyword extraction process

This method use the training corpus to train the keyword extraction model, and perform keyword extraction on the documents that need to be extracted according to the model.

### Semi-supervised keyword extraction algorithm

This kind of method requires only a small amount of training data. The algorithm also regards the keyword extraction as a binary classification problem, labeling the candidate words as keywords and non-keywords, and then use semi-supervised learning method to extract keywords from new texts, for example, the method of Support Vector Machine (SVM)[[26](#_2koq656)] based on semi-supervised learning and label propagation[[27](#_zu0gcz)].

### Unsupervised keyword extraction algorithm

The corpus that does not need to be manually labeled. The method is to first extract candidate words, then calculates the weight for each candidate word. Finally output the top-K candidate words with the highest weights as keywords. Since the supervised text keyword extraction algorithm re- quires high manual expenses, so most researches mainly use unsupervised keyword extraction with strong applicability to conduct experiment.

F Liu[[28](#_3jtnz0s)] and S Rose[[24](#_1ljsd9k)] proposed three methods for unsupervised keyword extraction methods. There are different algorithms according to different weight-calculating strategies, such as TF-IDF[[29](#_1yyy98l)], TextRank[[30](#_4iylrwe)], LDA[[31](#_2y3w247)], etc.

* + - 1. Keywords extraction based on statistical features (such as TF, TF-IDF). The idea of the keyword extraction algorithm based on statistical fea- tures is to extract the keywords of the document by using the statistical information of the words in the document;
      2. Keywords extraction based on word graph models (such as PageRank, TextRank). Keyword extraction based on the word graph model first builds the language network graph of the document, and then ana- lyzes the language network graph to find important words or phrases on this graph. These phrases are the keywords of the document;
      3. Keyword extraction based on topic models (such as LDA). The topic- based keyword extraction algorithm mainly uses the feature of the topic distribution in the topic model to conduct keyword extraction.

The unsupervised method does not require the process of manually labeling the training set, hence it is faster. But it cannot effectively use a variety of

information to rank candidate keywords, the effect of this method cannot be comparable to that of the supervised method. While the supervised method performs better, since it can adjust the influences that information has on accessing the keywords through training and learning[[25](#_45jfvxd)]. Supervised text keyword extraction algorithms require high labor costs.

## Topic model

The topic model[[32](#_1d96cc0)] has received more and more attention in NLP. In this field, the topic can be defined as the probability distribution of the term. The topic model extracts semantically related topics through the co-occurrence information of the term at the document level collection, and can transform the document in the term space to the topic space, and attain the transfor- mation of the document in the low-dimensional space.

Topic Model is a statistical model that clusters the latent semantic structure of documents with an unsupervised learning method.

The topic model defines that there is no direct connection between words and documents. They should also have a dimension to connect them. The topic model consider the dimension as a topic. Each document should corre- spond to one or more topics, and each topic will have a corresponding word distribution pattern. Through analyzing the topic, the word distribution of each document can be obtained. Based on this principle, a core formula of the topic model can be defined as follow:

*K*

*p*(*wi*|*dj*) = ∑ *p*(*wi*|*tk*) × *p*(*tk*|*dj*)

*k*=1

Where *wi* refers to a certain word in the document *dj*, and *tk* refers to the topic in the document *dj*. In a predefined data set, *p*(*wi*|*dj*) corresponding to each word and document. The topic model is based on the information

and calculates the values of *p*(*wi*|*tk*) and *p*(*tk*|*dj*) to obtain the word distri- bution of the topic and the document theme distribution information. To get this distribution information, methods such as Latent Semantic Indexing (LSI)[[33](#_3x8tuzt)] and Latent Dirichlet Allocation(LDA)[[34](#_2ce457m)] are the commonly used. Among them, LSI mainly uses the Singular Value Decomposition (SVD)[[35](#_rjefff)] method for brute force cracking, while LDA uses the Bayesian method[[36](#_3bj1y38)] to fit the distribution information.

### LDA

Latent Dirichlet Allocation (LDA) was proposed by David Blei[[34](#_2ce457m)] in 2003. The theoretical basis of this method is Bayesian theory[[37](#_1qoc8b1)]. Based on the analysis of the co-occurrence information of words, LDA fits the distribution of word-document-topic, and then maps both words and text into a semantic space. LDA is defined as a three-layer Bayesian probabilistic model, which contains a three-layer structure of words, topics, and documents; uses the co-occurrence relationship of words in documents to cluster words by topic to obtain "document-topic" and "topic-word" in a probabilistic distribution. A three-layer Bayesian model is presented as follows in figure[2.4](#_49x2ik5):

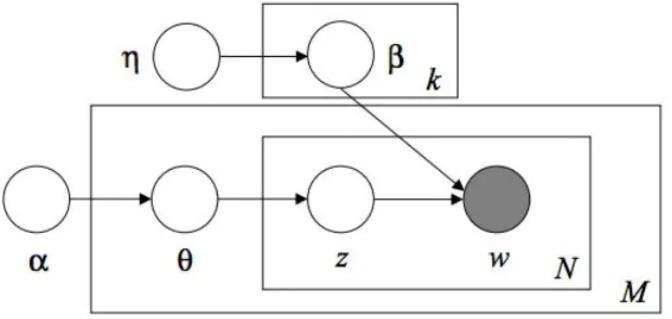


Figure 2.4: A three-layer Bayesian model

LDA method assumes that the prior distribution of the topic in the doc- ument and the prior distribution of the words in the topic all fulfill the Dirichlet distribution. From the perspective of Bayesian method, posterior distribution refers to prior distribution with data. The polynomial distri- bution of the topic in each document and the polynomial distribution of the corresponding word of each topic can be obtained through the statistics of the existing data set. According to the Bayesian method, through the prior Dirichlet distribution and the polynomial distribution obtained from the observation of the data, a set of Dirichlet-multi conjugates[[38](#_4anzqyu)] can be obtained, and the posterior distribution of the topics in the document can be inferred based on this. The final outcome can be calculated through this approach. According to most of related studies, one of the main method to obtain LDA model is Gibbs sampling[[39](#_2pta16n)]. The LDA model training process combined with Gibbs sampling is generally as follows:

* + - 1. Random initialization, for each word w in each document in the corpus, assigning a topic number randomly.
      2. Rescan the corpus, for each word w, resample its topic according to the Gibbs sampling formula, and update it in the corpus.
      3. Repeat the above corpus resampling process until Gibbs sampling converges.
      4. The topic-word co-occurrence frequency matrix of the statistical cor- pus. This matrix is the model of LDA.

After the above steps, a trained LDA model is obtained, and then the topic of the new document can be estimated in a certain way. The specific steps are described as follows:

1. Random initialization, for each word w in the current document, a topic number z is randomly assigned.
2. Rescan the current document and resample its topic according to the Gibbs sampling formula.
3. Repeat the above process until Gibbs sampling converges.
4. The topic distribution in the statistical file is the estimated result.

### LSI

Latent Semantic Indexing (LSI) is based on the SVD to get the topic of the certain document. SVD is an algorithm widely utilized in machine learning. It can be applied not only in feature decomposition in dimension reduc- tion algorithms, but also for recommendation systems, natural language processing and other fields. It is also the cornerstone of most of machine learning algorithms.

## Text generation

### General information

As we mentioned before, the main task of the natural language processing model is to perform a series of processing on natural language to achieve the user’s purpose. Compared with machine language, natural language is not restricted in a particularly strict format, therefore, it is considerably difficult to analyze and generate text from the sentence structure. So the current text generation is achieved through the connection between words, as the prompt of the input method, which the next word is generated after inputting a word. These steps are the basic process of the current text generation model. A list of words with the most relevance to the previous word and one of them selected as output, and generate an article in repeated steps. Figure[2.5](#_1hmsyys)[[40](#_14ykbeg)] presents a process of text generation with Recurrent Neural Network(RNN)[[9](#_haapch)].

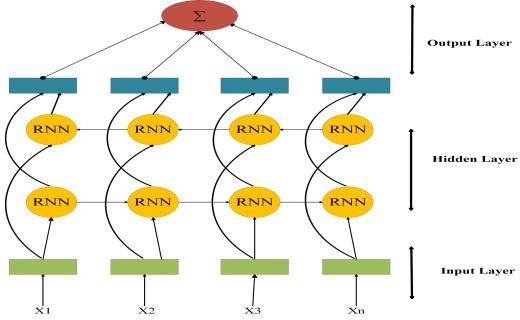


Figure 2.5: Text generation with Recurrent neural network

Text generation can be implemented through a deep neural network struc- ture. The Sequence-to-Sequence model[[41](#_3oy7u29)] proposed by Google Brain team in 2014 brought the end-to-end network into the research of NLP. Sequence- to-Sequence is also known as the encoder-decoder (Encoder, Decoder) struc- ture. Among them, Encoder and Decoder are usually composed of several layers of RNN. Encoder is designed for encoding the original text into a vector while decoder is designed for extracting information from the vector to obtain semantics and generate text[[41](#_3oy7u29)].

A review by Z, Yu[[42](#_243i4a2)] summarized the advantage of end-to-end framework is that by reducing manual preprocessing and subsequent processing, the model is as far as possible from the original input to the final output, pro- viding the model with more resource for automatic adjustment based on the data, and increasing the overall fitting ability of the model.

Concerning to the encoder part, there are several types that have been widely used[[42](#_243i4a2)]: Bag-Of-Words Encoder[[43](#_j8sehv)], RNN Encoder[[44](#_338fx5o)] and Bidi- rectional Encoder[[45](#_1idq7dh)]. Bag-of-words is a representation of the text that

describes the occurrence of words in a document. It was originally used in text classification to represent documents as vectors. This method is to assume that for a text, its word sequence, grammar, and syntax are ignored, and it is only regarded as a collection of vocabulary, and each word in the text is independent. So bag-of-words encoder averages word embedding, regardless of the sequence of words and the connection between words. Fig- ure[2.6](#_41mghml) presents an example of bag-of-words model. In this example, every single words in the documents were counted independently and recorded in the table.

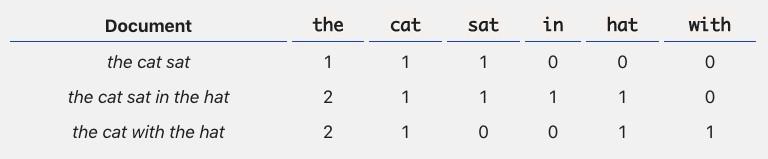


Figure 2.6: A example of bag-of-words model

RNN encoder exists the problem of gradient disappearance and gradient explosion. Bidirectional encoder learns the hidden state from the front and back directions. As for the decoder, the Neural Network Language Model, RNN Decoder and Attention-Based RNN Decoder are mainly used. The Neural Network Language Model does not consider the contribution of historical information to the generated text, so the attention mechanism was added later.

As for the training strategy, there are mainly two training strategies, one is the word-level maximum probabilistic estimation method (Maximum probabilistic Estimation), and the other is the sentence-level minimum risk training method (Minimum Risk Training)[[42](#_243i4a2)].

### Language model

The function of the language model is to predict the next word based on the previous sentence. Pre-training language model in nlp mainly includes three aspects: Auto-regressive pre-training language models represented by unidirectional features, also called as unidirectional models, for instance, GPT2; self-encoding pre-training language model of bidirectional feature representation, also called BERT model, for instance, BERT; auto-regressive pre-training language model with bidirectional feature representation: XL- Net;

GPT-2 is one of the excellent applications of machine learning. It has amaz- ing performance in text generation, and the generated text exceeds people’s expectations of what the current language model produce in terms of con- textual coherence and emotional expression. The GPT-2 was a considerably huge transformer-based language model trained on a large dataset. Bidirec- tional Encoder Representation from Transformers(BERT), is a pre-trained language model. It emphasizes that the traditional one-way language model or the shallow splicing method of two one-way language models are no longer used for pre-training as in the past, but a new masked language model (MLM) is used to generate deep Two-way language representation. The following section introduces GPT-2 briefly.

### Transformer

Transformer is proposed by the paper "Attention is All You Need"[[10](#_319y80a)], which reduces the training time of RNNs and uses self-attention mechanism to achieve fast parallelism. Transformer is an end-to-end Seq2Seq structure, which contains two components: Encoder and Decoder. Encoder consists of 6 identical layers, and each layer is composed of two sub-layers, which are a multi-head self-attention mechanism and a fully connected feed-forward network. Each sub-layer has added residual connection and normalisation.

The structure of Decoder is similar with Encoder, but there is an additional sub-layer of attention,

### GPT-2

The construct of GPT-2 is based on the "transformer decoder module", while BERT is built using the "transformer encoder". One thing should be pointed out is that one of the key differences between the two is that GPT-2 only outputs one word (token) at a time, like a traditional language model.

The reason why this model performs better than other traditional model is that after a new word is generated, the word is added after the previously generated word sequence, which will become the new input for the next step of the model. This type of mechanism is auto-regression, which is also an important method that makes the RNN model more effective.

GPT-2, as well as some subsequent models such as TransformerXL and XLNet, are essentially auto-regressive models, while BERT is not. This results in a kind of trade-off problem. Although without employing auto- regressive mechanism, BERT has obtained the ability to combine contextual information, thus achieving better results. XLNet uses auto-regression and introduces a method that can take into account the contextual information.

## Terms and concepts

TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. Words that are common in a single or a small group of documents tend to achieve higher TF-IDF numbers than

common words such as articles and prepositions.

TF (Term Frequency, abbreviated as TF)[[46](#_42ddq1a)] refers to frequency of the word

, that is, the number of total amounts a word presents in the text, and the count is the word frequency TF. Apparently, if a word appears many times in the article, this word would possess a significant effect. But if we practice it ourselves, we will definitely find out that the TF counted are mostly words like:’is’,’Yes’. These words do not contribute to our analysis and statistics, but sometimes it will interfere with our statistics. Therefore, we need to remove these worthless words. There are a large amount of approaches to remove these words, such as using a corpus of stop words.

Usually, TF value will be normalized (generally the word frequency will be divided by the total amounts of words in the article) to prevent it from being biased towards longer documents. The same word may have a higher word frequency in a long document than in a short document, regardless of the importance of the word.

However, one thing should be noted is that some common words do not have much effect on the topic. On the contrary, some words that appear less frequently can express and represent the main topic of the article, so it is not appropriate to use TF only. Weight design requires that if the abil- ity of a word to predict the topic is strong enough, the weight should be considerable great corresponds to that, and vice versa. In all articles that needed to be accounted, some words only appear in a few of them, so such words have a great effect on the topic of the article, and the weight of these words should be designed to be larger. For instance, assuming there is an article "Bee Farming in China", the most frequently used words are "’s", "is", "at", which are "stop words". So these words do not contribute in analyzing and should be removed. Words such as "Bees" and "farming" are not so common but are important, so these words are rare but useful and their

weights should be designed larger.

Generally speaking, a vector space model transforms the query keywords and documents to vectors, and wield the calculations between the vectors to further represent the relationship between the vectors. For instance, a com- monly used calculating operation is to calculate the "correlation score"[[47](#_2hio093)] between the vector corresponding to the query keyword and the vector corresponding to the document.

The implicit assumption behind TF is that the words in the query retrieval keyword should be more important than other words, and the importance of the document, that is, the degree of relevance, is proportional to the number of times the word appears in the document. For instance, if the word "Car" appears 5 times in document A and 20 times in document B, then the TF calculation considers document B to be more relevant.

Another approach is Inverse Document Frequency (IDF), which refers to the frequency of a word in all texts (the entire document). If a word presents in most of texts, the IDF value of it should be low. On the other hand, if a word appears in relatively few documents, the IDF value of it should be high. For example, some professional terms such as "Machine Learning". The IDF value of such words should be large. In an extreme case, if a word appears in all texts, its IDF value should be zero.

When we calculate the importance of a word, if use TF or IDF respectively, the results would be one-sided. Thus, TF-IDF combines the advantages of both TF and IDF to evaluate the importance of a word for a document set or one file in a corpus. The importance of a word increases in proportion to the number of times it appears in the document, but at the same time it decreases in inverse proportion to the frequency of its appearance in the corpus. Therefore, the more a word appears in an article, and the less it ap-

pears in all documents, the more it can represent the article and distinguish it from other articles.

TF refers to the word frequency after normalization, and IDF refers to the in- verse document frequency. Given a document set D, there are *d*1, *d*2, *d*3, , *dn* ∈

*D*. Generally, stop words such as "is" are removed when calculating TF-IDF.

The document set contains a total of m words, including *w*1, *w*2, *w*3, , *wm* ∈

*W*. We set the calculation of the TF-IDF of the word *wi* in the document *dj*

as an example. The calculation formula of TF is:

freq(*i*,*j*)

*TF* =

maxlen (j)

where *f req*(*i*, *j*)is the frequency of *wi* in *dj*, and maxlen (*j*) is the length of *dj*. TF can only describe the frequency of the word in the document, but assum- ing there is a word "we", this word may appear in every document in the document set D, and has a higher frequency. Then this type of word does not have a proper ability to distinguish documents. In order to reduce the effect of this common word, the idea of IDF[[29](#_1yyy98l)] is introduced. The formula of IDF is shown as follows:

*IDF* = log len(*D*)

*n*(*i*)

where *len*(*D*) represents the total number of documents in the document set D, and *n*(*i*) represents the number of documents that contain the word *wi*.

After getting TF and IDF, we multiply these two values to get the value of TF − IDF:

TF − IDF = TF ∗ IDF

TF calculates the frequency of words in a document, and IDF can reduce the effect of some common words. Therefore, for a document, we can use the vector of TF-IDF of each word in the document to represent the document, and then calculate the correlation between documents based on methods

such as cosine similarity[[29](#_1yyy98l)].

# Method, system design and development

This section describes the method and design of the system in detail. In keyword extraction module, we use the topic model and two strategies: LDA and LSI to obtain the keywords. And for story generation module, we build the module based on the pre-trained model GPT-2.

## Dataset and data preprocessing

This part of the work mainly includes:

Collect short story text sets for training and testing, and divide the training test set appropriately (usually, the ratio of the training set text and the extra test set should be 7:3, 70 percents of the text is used for training, and the remaining 30 percents is used for testing evaluation algorithm effects). In this project, we choose to use the Edmonton Narrative Norms Instrument (ENNI) database[[48](#_wnyagw)]. ENNI database is an open source storytelling database that collects structured language production samples from children aged 4 to 9 through storytelling activities. The ENNI consists of two sets of three- story picture books from which children tell a total of six stories. It fits the requirements of our project the best and is an ideal dataset.

The data set need to be pre-processed to make it suitable for the deep learn- ing model. The original database from the ENNI are recorded (video- or audio-taped) and transcribed later by using a simple word processing pro- gram. Each original file has certain information in it and they must begin with certain two lines which consists the participants information. Since these data sets are collected from transcripts, there are some transcription conventions as shown in the table[3.1](#_46r0co2).

|  |  |
| --- | --- |
| Example | Information |
| *"@Begin" "@Languages: eng"*  *"@Participants: CHI Target-Child, EXA Investigator"*  *"@Media: 704, audio, unlinked" "@Comment: Birth of CHI is 28-DEC- 1992"*  *"@Date: 09-MAR-2000"*  *"@Tape Location: Disk M3 Track 55" "@G: A1"*  *"@Types: cross, narrative, TD"* | Each file must begin with the follow- ing two lines:  @BEGIN  @PARTICIPANTS: CHI-target child, EXA-examiner;  The PARTICIPANTS line tells the program who the transcript lines belong to. In addition, there are other lines contained information like recording date, location, etc. |

Table 3.1: Original text and contained information

So in this case, we need to remove the noise in the data set, including mean- ingless modal particle, non-text content, etc. Compared with traditional text classification, the noise problem of short story text classification data is more severe, for the reason that under the condition of short text, there are fewer features in each story, and noise will have a considerable impact on the final classification result. Therefore, removing noise data becomes a vital and meaningful task.

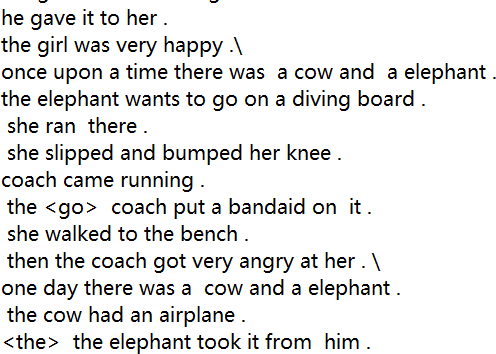


Figure 3.1: Example of separate stories

For these short story text, we also need to remove stop-words, for exam- ple,"*’s*" "*’t*", etc, and detain stemming. Since the keyword extraction module will take each single story as input, we need to separate these story by separable character, in this case, we use "\" to separate these stories, shown as follow:

## Text vectorization

Similar to the traditional text categorization task, we need to manage the computer interpreting the text of natural language contents. To do so, we choose to take certain measures to vectorize these short texts, which mainly includes the following steps:

* + 1. Document Indexing. Selecting the appropriate feature unit to in- dex the text (document indexing). Document Indexing also plays an important role in other fields such as information retrieval. In text categorization, the Bag-of-words Model (BOW)[[49](#_3gnlt4p)] is usually applied, which assumes that each feature unit[[50](#_1vsw3ci)] corresponds to one dimen-

sion in the feature space[[51](#_4fsjm0b)], so that the text can be denoted as a word frequency vector in the feature space.

* + 1. Weight calculation. From an intuitive perspective, the importance of each feature unit in the article is different. Generally speaking, the higher the frequency is, the greater the weight of the feature unit has. At the same time, we also need to take the noise impact caused by stop words into account. From the perspective of research, the most commonly used weight calculation method is the tf\*idf weight calculation method[[46](#_42ddq1a)].
    2. Feature selection. Due to the particularity of natural language, in tra- ditional text categorization tasks, the feature space used for document indexing generally has a higher dimension, which will considerably increase the training cost when training the classifier model. In addi- tion, Some of the noise features contained also have a negative effect on the performance of the final classifier. Therefore, it is necessary for us to select the features in the original space to obtain a more effective feature subspace. However, in short text categorization tasks, the problem of feature sparseness is notably crucial, so we should be thoughtful when selecting features.

Through the above steps, we manage transforming the text into a form that the computer can perceive. What needs to be pointed out is that for the text in the training set and the test set, we must use the same method to process these data, that is: the feature unit, feature selection method and weight calculation must be the same, so that it can be performed studying and testing under the same classifier model.

## System development

The structure of the system is shown in the figure [3.2](#_2zbgiuw) blow:

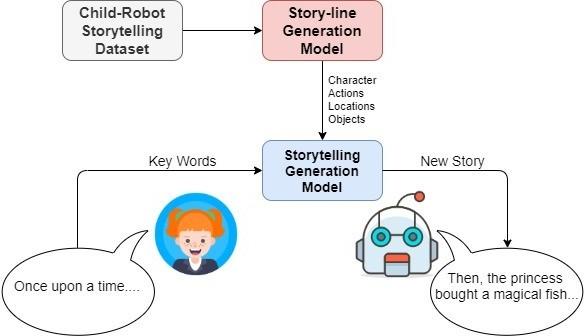


Figure 3.2: The architecture of the storytelling system

First we extract the stories told by children and create a storytelling corpus that would be integrated with the Open Source storytelling database ENNI. And this corpus will be used in the next step.

Then we will conduct data processing for natural language processing. In this step, we choose to use topic model to extract the keywords of each story. We used LDA and LSI in topic model.

LDA obtains the probability distribution topic of each document. After analyzing the documents to extract their topics distribution, it clusters the topics according to the topics distribution. In addition, a document usually contains multiple topics, and each word in the document is generated from one of the topics. While LSI uses the results to build relevant indexes after analyzing, it performs singular value decomposition (SVD) operation on the word-document matrix, and maps the word-document matrix to a lower-

dimensional approximate SVD according to the SVD result. Each word and documents can be represented as a point in the topics space. By computing the similarity between each word and the document, we can obtain the similarity result for each word in each document. And we choose the top-4 words with the highest similarity to be keywords fot the document. Since LDA and LSI both analyze the latent semantics of the document, we can consider the two algorithms to be the same.

We build the generation module based on pre-trained models, GPT-2[[52](#_2uxtw84)]. The model will take the keywords extracted from the story to generate a new story based on the new storytelling corpus.

GPT-2 collects the embedding vector corresponding to the word from the embedding matrix, which is also part of training result of the model.Each row in document is a word embedding vector, which means a list of num- bers that can represent a word and obtain the meaning. The length of the embedding vector is related to the size of the GPT-2 model. The smallest model uses an embedding vector with a length of 768 to represent a word. Before entering a word into the first transformer module, we need to find its corresponding embedding vector, adding the position vector corresponding to position number one.

The steps of the first transformer module to process words are as follows: first process them through the self-attention layer, and then pass them to the neural network layer. After the first transformer module processed, the result vector is passed to the next transformer module in the stack to con- tinue the calculation. The processing method of each transformer module is the same, but each module maintains its own self-attention layer and weights in the neural network layer. Self-attention mechanism integrates the model’s understanding of related words used to represent the context of a word before processing each word. The specific method is to assign

a relevance score to each word in the sequence, and then sum their vector representations. After the last transformer module outputs the result, the model will multiply the output vector by the embedding matrix. In this way, the model completes a round of iteration and outputs a word. The model will continue to iterate until a complete sequence is generated-the sequence reaches the upper limitation of 1024-length or a terminator character is generated in the sequence.

# Results and evaluation

This section presents part of the output of the two module respectively and also describes automated method for evaluating the performance of the story generation model, and human evaluation to evaluate to what extent the story generated perceived by human.

## Results

### Keyword extraction

In this process, the module filters the words in the stop-word list and words of length under 2-gram. We perform a weighted calculation according to tf- idf for each word, to obtain a weighted vector representation, and calculate the similarity between the distribution of the word and the distribution of the document, and take the pre-defined number of words with the highest similarity as keywords outcome. Part of the results is presented in table [4.1](#_1rvwp1q). Based on current observation, the number of topics in most topic models is

|  |  |
| --- | --- |
| Sentences as input | *"one day there was an elephant and a giraffe who wanted*  *to go the elephant dropped her ball in the swimming pool. the giraffe swam and got it for her. he gave it to her. the girl was very happy. once upon a time there was a cow and a elephant . the elephant wants to go on a diving board. she ran there. she slipped and bumped her knee. coach came running. the coach put a bandaid on it. she walked to the bench. then the coach got very angry at her."* |
| Keyword as output | elephant, pool, coach |

Table 4.1: A sample output of Keyword extraction module

generally selected based on experience. The Hierarchical Dirichlet Process (HDP)[[53](#_1a346fx)] select the number of topics automatically. One problem of the LSI method is that the outcome of LSI achieved is not a probabilistic model. It lacks the statistical foundation, and the results are difficult to interpret intuitively. Regarding this problem, Hofmann[[54](#_3u2rp3q)] proposes topic models based on probabilistic distributions such as probabilistic latent semantic indexing (also inferred as pLSA) and Latent Dirichlet distribution (LDA) to replace the topic models based on matrix factorization.

From the perspective of application, as for some small-scale problems, if the purpose is to find out some topic distribution relationships quickly and coarse-grained, LSI is a thoughtful choice. At for other purpose, choosing LDA and HDP would be a better choice.

### Story generation

Part of the results is presented below in table [4.2](#_1664s55).

|  |  |
| --- | --- |
| Sentences as input | *"one day there was an elephant and a giraffe who wanted to go the elephant dropped her ball in the swimming pool. the giraffe swam*  *and got it for her. he gave it to her. the girl was very happy."* |
| Story as output | and then he was trying to get it. <and then he got> and then the elephant wanted to and then <um> the giraffe was sad and then the lifeguard came. <and h> and the elephant lifeguard gave <him> him some money. and then they gave  him some money. and then the giraffe was happy. the end. |

Table 4.2: A sample output of text generation module

## Evaluation method

### Automated metrics

Since our task is to generate a new story from the given text, most of the common nlp evaluation methods are not suitable for evaluating the output. For instance, the BLUE score[[55](#_2981zbj)] is a method for automatic evaluation of machine translation, it represents how similar between the candidate text and the reference text. In this case, we need to evaluate a new story from different aspect, which means comparing how similar the generated text and the original text is meaningless.

The common evaluation method for evaluating the effectiveness of lan- guage models is perplexity. The lower the perplexity obtained on a test set, the better the modeling effect. The formula for calculating perplexity is

described as follows:

*perplexity*(*S*) = *p* (*w*1, *w*2, *w*3, . . . , *wm*)−1/*m*

s*m m*  1

*i*=1

= ∏ *p* (*wi* | *w*1, *w*2, . . . , *wi*

−1)

where the *m*-length sentence *s* is regarded as a sequence of words {*w*1, *w*2, . . . , *wm*}, and *p* (*w*1, *w*2, . . . , *wm*) refers to the probability distribution to be modeled.

To put it simply, perplexity describes the ability of a language model to predict a language data set. For instance, if a known sentence will appear in the corpus, the higher the probability of the sentence calculated by the language model, the better the model fits this corpus. In the training process of language models, the logarithmic expression of perplexity is usually used:

log

perplexity (*S*

1 *m p w*

*w* , *w* , . . . , *w*

( )) = ∑

— *m*

*i*=1

( *i* | 1 2

*i*−1)

Compared with the method of finding the square root of the product, the form of addition can accelerate the calculation, and at the mean time avoid- ing the problem of the floating-point number overflow caused by too small probability product value.

Mathematically, log perplexity can be regarded as the cross entropy between the real distribution and the predicted distribution. Cross Entropy describes a type of distance between two probability distributions. The formula is presented as follow:

*H*(*u*, *v*) = *Eu*[−log *v*(*x*)] = − ∑ *u*(*x*) log *v*(*x*)

*x*

There are some disadvantages of perplexity. When the data set is consider- ably large, the perplexity decreases fast. The non-textual character in the

data posses a great impact on the perplexity of the model. And common words also interferes on it. The perplexity of our text generation model is 3.24, which is considerably low and means a good predict ability on the data set.

### Human evaluation

In the evaluation phase, we address the following research question: How are the stories perceived by human participants in terms of cohesion and co- herence. To address this, we decide to use crowd-sourced workers method. The survey is conducted on Amazon Mechanical Turk. It relies on the an- swer replied by workers, which is not always truthful, since worker would pick the same option for all questions[[56](#_odc9jc)].

The questionnaire consists of 4 stories that are selected randomly from the output of the model. Each story is followed by 3 questions based on the story. For each story, there are two evaluations: cohesion and coherence. Cohesion refers to how well the two consecutive sentences are locally con- nected and coherence measures to what extent multiple sentences globally hold together[[57](#_38czs75)]. For each question, there are 5 scales. We also set an atten- tion check question for each part to prevent worker from picking random answer. If the participants passed the attention checks, we consider that this data is valid. If more than 70% answers are correct, we then consider the answers to be truthful. According to the results, the lowest percentage of correct answer is 73.53%, so we consider these survey results as truthful.

We recruited 34 workers participated in our survey. The survey also in- cluded demographics information, about 80% workers are fluent or native English users. In the coherence evaluation, if the percentage of workers that pick options medium, good or very good higher than 50%, we then consider this story has good coherence. According to the results, the lowest and the

highest percentage of workers who evaluated the text as medium, good or very good are 55.88% and 79.41% respectively and the average is 64.71%.

And in the cohesion evaluation, if the percentage of workers that pick options medium, good or very good higher than 50%, we then consider that this story has good cohesion. According to the results, the lowest and the highest percentage of workers who evaluated the text as medium, good or very good are 44.18% and 82.35% respectively and the average percentage is 61.76%. Our results suggest that the stories generated by our system are well perceived by workers and also have acceptable cohesion and coherence. The results are presented as below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | The lowest  percentage(%) | The highest  percentage(%) | Average  percentage(%) |
| Coherence | 55.88 | 79.41 | 64.71 |
| Cohesion | 44.18 | 82.35 | 61.76 |

Table 4.3: Survey results

# Discussion and Conclusions

## Discussion

In keyword extraction module, rules of selecting certain keywords can be improved. Evaluating the final results from different dimensions of at- tributes and through different methods are worth investigating.

In our experiment, the corpus consists of about 300 stories, and the time expense of story generation is high, which is almost 30 seconds. So the time complexity is considerably huge. Especially when it comes to a real application scenario, the corpus would be huge, so the effectiveness of the system needs to be improved to reduce the time complexity.

As we mentioned before, the story is generated from the topic-word corpus that is created from the prior step, if the topic that is extracted from the input sentences is not included in that corpus, the generation module would generate the story randomly, which is out of our expectation. To address this problem, we need to expand our database to include more topics or relate the new topic to the topic that already exists in the corpus.

From the view point of application, the system can further test the appli- cability on the robot, whether it can achieve a part of the function of a good interactive educational robot, in order to carry out more practical tests. It will be of great significance to study the theoretical application and

improvement of this project.

## Conclusion

The system consists of two components, including a keyword extraction module, which uses sentences collected from children as input to extract keywords as output, and a text generation module, which uses keywords that are extracted from the previous module from the corpus to gener- ate a new story, where natural language processing and machine learning techniques are applied in. Therefore, the contribution of this project in theoretical research is reflected in the exploration, discussion and research of these two technologies.

This kind of interactive communication that the system provide is consid- erably helpful for improving children’s recognition and communication abilities. It is an crucial aspect of advancing personalized education, im- proving the tutoring and interaction approach, and is also the foundation of future education. The system takes sentences collected from children as input, extracting keywords through the topic model, and the text generator which is implemented with GPT-2 takes keywords from previous model as input and generates a new story from the corpus that is generated from keyword extraction module.

Considering that evaluation metrics as perplexity do not provide enough in- formation to draw strong conclusions regarding the accuracy and efficiency of the system. Therefore, we included metrics of cohesion and coherence using co-workers. From the evaluation method, we can conclude that our output story can be well perceived by human.

# Bibliography

* + 1. L. P. E. Toh, A. Causo, P.-W. Tzuo, I.-M. Chen, and S. H. Yeo, “A review on the use of robots in education and young children,” *Journal of Educational Technology & Society*, vol. 19, no. 2, pp. 148–163, 2016.
    2. T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and

F. Tanaka, “Social robots for education: A review,” *Science Robotics*, vol. 3, no. 21, 2018. eprint: <https://robotics.sciencemag.org/content/3/21/eaat5954.full.pdf>.

* + 1. F. B. V. Benitti, “Exploring the educational potential of robotics in schools: A systematic review,” *Computers & Education*, vol. 58, no. 3, pp. 978–988, 2012.
    2. J. McCarthy, “What is artificial intelligence?,” 1998.
    3. W. Song and J. Cai, “End-to-end deep neural network for automatic speech recognition,” *Standford CS224D Reports*, 2015.
    4. M. S. Hossain and G. Muhammad, “Emotion recognition using deep learning approach from audio–visual emotional big data,” *Information Fusion*, vol. 49, pp. 69–78, 2019.
    5. L. Yao, N. Peng, R. Weischedel, K. Knight, D. Zhao, and R. Yan, “Plan-and-write: Towards better automatic storytelling,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7378–7385.
    6. S. Hochreiter and J. Schmidhuber, “Long short-term memory,”

*Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

* + 1. T. Mikolov, M. Karafiát, L. Burget, J. Cˇ ernock, and S. Khudanpur, “Recurrent neural network based language model,” in *Eleventh annual conference of the international speech communication association*, 2010.
    2. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones,

A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017.

* + 1. E. D. Liddy, “Natural language processing,” 2001.
    2. P. Jackson and I. Moulinier, *Natural language processing for online applications: Text retrieval, extraction and categorization*. John Benjamins Publishing, 2007, vol. 5.
    3. A. Kao and S. R. Poteet, *Natural language processing and text mining*. Springer Science & Business Media, 2007.
    4. G. G. Chowdhury, “Natural language processing,” *Annual review of information science and technology*, vol. 37, no. 1, pp. 51–89, 2003.
    5. X.-T. Xiao and S.-I. Kim, “A study on the user experience of smart speaker in china-focused on tmall genie and mi ai speaker,” *Journal of Digital Convergence*, vol. 16, no. 10, pp. 409–414, 2018.
    6. G. López, L. Quesada, and L. A. Guerrero, “Alexa vs. siri vs. cortana vs. google assistant: A comparison of speech-based natural user interfaces,” in *International Conference on Applied Human Factors and Ergonomics*, Springer, 2017, pp. 241–250.
    7. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
    8. D. W. Otter, J. R. Medina, and J. K. Kalita, “A survey of the usages of deep learning for natural language processing,” *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
    9. T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *CoRR*,

vol. abs/1708.02709, 2017. arXiv: [1708.02709](https://arxiv.org/abs/1708.02709).

* + 1. R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and

A. Jatowt, “A text feature based automatic keyword extraction method for single documents,” in *European conference on information retrieval*, Springer, 2018, pp. 684–691.

* + 1. Wikipedia contributors, *Baidu — Wikipedia, the free encyclopedia*, [Online; accessed 17-March-2021], 2021.
    2. ——, *Microsoft bing — Wikipedia, the free encyclopedia*, [Online; accessed 17-March-2021], 2021.
    3. P. Magnani, *Understanding what’s behind the lyrics*, 2020.
    4. S. Rose, D. Engel, N. Cramer, and W. Cowley, “Automatic keyword extraction from individual documents,” *Text mining: applications and theory*, vol. 1, pp. 1–20, 2010.
    5. S. Beliga, “Keyword extraction: A review of methods and approaches,” *University of Rijeka, Department of Informatics, Rijeka*, pp. 1–9, 2014.
    6. K. Bennett, A. Demiriz, *et al.*, “Semi-supervised support vector machines,” *Advances in Neural Information processing systems*, pp. 368–374, 1999.
    7. B. Cui, X. Xie, S. Hao, J. Cui, and Y. Lu, “Semi-supervised classification of hyperspectral images based on extended label

propagation and rolling guidance filtering,” *Remote Sensing*, vol. 10, no. 4, p. 515, 2018.

* + 1. F. Liu, D. Pennell, F. Liu, and Y. Liu, “Unsupervised approaches for automatic keyword extraction using meeting transcripts,” in *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, 2009, pp. 620–628.
    2. W. Zhang, T. Yoshida, and X. Tang, “A comparative study of tf\* idf, lsi and multi-words for text classification,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011.
    3. R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.
    4. H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, “Latent dirichlet allocation (lda) and topic modeling: Models, applications, a survey,” *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15 169–15 211, 2019.
    5. X. Yan, J. Guo, Y. Lan, and X. Cheng, “A biterm topic model for short texts,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 1445–1456.
    6. D. A. Grossman and O. Frieder, *Information retrieval: Algorithms and heuristics*. Springer Science & Business Media, 2012, vol. 15.
    7. D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,”

*the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

* + 1. V. Klema and A. Laub, “The singular value decomposition: Its computation and some applications,” *IEEE Transactions on automatic control*, vol. 25, no. 2, pp. 164–176, 1980.
    2. M. Lavine and M. West, “A bayesian method for classification and discrimination,” *Canadian Journal of Statistics*, vol. 20, no. 4,

pp. 451–461, 1992.

* + 1. J. K. Ghosh, M. Delampady, and T. Samanta, *An introduction to Bayesian analysis: theory and methods*. Springer Science & Business Media, 2007.
    2. X. Wu, H. Wu, X. Xie, J. Xu, and T. Zhang, “Short text topic recognition and optimization method for university online community,” in *International Conference on Artificial Intelligence and Security*, Springer, 2019, pp. 79–90.
    3. A. E. Gelfand, “Gibbs sampling,” *Journal of the American statistical Association*, vol. 95, no. 452, pp. 1300–1304, 2000.
    4. I. Milanova, K. Sarvanoska, V. Srbinoski, and H. Gjoreski, “Automatic text generation in macedonian using recurrent neural networks,” in *International Conference on ICT Innovations*, Springer, 2019, pp. 1–12.
    5. K. Cho, B. Van Merriënboer, 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.

* + 1. S.-Q. Shen, Y.-K. Lin, C.-C. Tu, Y. Zhao, Z.-Y. Liu, M.-S. Sun, *et al.*, “Recent advances on neural headline generation,” *Journal of computer science and technology*, vol. 32, no. 4, pp. 768–784, 2017.
    2. Y. Fu, Y. Feng, and J. P. Cunningham, “Paraphrase generation with latent bag of words,” *arXiv preprint arXiv:2001.01941*, 2020.
    3. V.-K. Tran and L.-M. Nguyen, “Natural language generation for spoken dialogue system using rnn encoder-decoder networks,” *arXiv preprint arXiv:1706.00139*, 2017.
    4. F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.
    5. Wikipedia contributors, *Tf–idf — Wikipedia, the free encyclopedia*, [Online; accessed 19-February-2021], 2021.
    6. J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson correlation coefficient,” in *Noise reduction in speech processing*, Springer, 2009, pp. 1–4.
    7. P. Schneider, R. V. Dubé, and D. Hayward, *The edmonton narrative norms instrument*, 2005.
    8. Wikipedia contributors, *Bag-of-words model — Wikipedia, the free encyclopedia*, [Online; accessed 19-February-2021], 2021.
    9. S. Zhang, Y. Qin, W.-J. Hou, and X. Wang, “Word segmentation and named entity recognition for sighan bakeoff3,” in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006,

pp. 158–161.

* + 1. B. Scholkopf, S. Mika, C. J. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A. J. Smola, “Input space versus feature space in kernel-based methods,” *IEEE transactions on neural networks*, vol. 10, no. 5,

pp. 1000–1017, 1999.

* + 1. P. Budzianowski and I. Vulic´, “Hello, it’s gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems,” *arXiv preprint arXiv:1907.05774*, 2019.
    2. Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical dirichlet processes,” *Journal of the american statistical association*, vol. 101, no. 476, pp. 1566–1581, 2006.
    3. T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’99, Berkeley, California, USA: Association for Computing Machinery, 1999,

pp. 50–57.

* + 1. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
    2. G. Paolacci, J. Chandler, and P. G. Ipeirotis, “Running experiments on amazon mechanical turk,” *Judgment and Decision making*, vol. 5, no. 5, pp. 411–419, 2010.
    3. W. Bublitz, “Cohesion and coherence,” *Discursive Pragmatics, John Benjamins Publishing Company, Amsterdam/Philadelphia*, pp. 37–49, 2011.