A Sentiment Analysis Method to Better Utilize User Profile and Product Information

A thesis submitted to the Department of Computing
of The Hong Kong Polytechnic University
in partial fulfillment of the requirements
for the degree of Bachelor of Science in Computing

Mingyu Derek MA
derek.ma@
Supervisor: Prof. Qin LU
Co-examiner: Dr. Ajay Kumar PATHAK
2nd Assessor: Dr. Richard LUI

July 1, 2018

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學
Department of Computing
電子計算學系

Abstract

Sentiment analysis which aims to predict users' opinions is a huge need for many industrial services. In recent years, many methods based on neural network achieve great performance on sentiment analysis such as Tang et al. (2014); Socher et al. (2013); Mikolov et al. (2013b). However, most of existing methods only focus on local documents and do not consider related user profile and product information (Kim, 2014; Yang et al., 2016; Long et al., 2017). Even some works attempt to combine those background information in sentiment analysis, they normally treat user profile and product information as a united entity and ignore that different background information may cause different aspects of influences to users' sentiments(Tang et al., 2015b; Chen et al., 2016; Dou, 2017). To address these issues, we proposed a new *Joint User and Product Memory Network (JUPMN)* utilizing user profile and product information in separate ways into sentiment classification. Inspired by the successful utilization of memory network (Weston et al., 2014; Sukhbaatar et al., 2015), our model first creates document representations using hierarchical LSTM model and then feeds the document vectors into new carefully designed user and product memory networks to reflect corresponding features. The evaluation of JUPMN on three benchmark review datasets IMDB, Yelp13 and Yelp14 shows that JUPMN outperforms the state-of-the-art model and further analysis of experimental results is employed.

Acknowledgements

I would like to convey my sincere gratitude to my supervisor, Prof. Qin LU[1], for her guidance and support. Her insights and knowledge led me to enter the field of Natural Language Processing, and her suggestions put me on the right track.

I am fortunate enough to learn from the faculties who helped me a lot during my undergraduate study and research. They are Prof. Jiannong CAO[1], Prof. Marilyn WALKER[2], Dr. Henry CHAN[1], Dr. Abel SANCHEZ[3], Prof. John R. WILLIAMS[3], Dr. Maggie Wenjie LI[1], Prof. Chu-Ren HUANG[4], Dr. Yao YAO[4] and Dr. Grace NGAI[1].

I am very grateful to get the support from Mr. Yunfei LONG[1], Mr. Rong XIANG[1] for this project, and Dr. Linchuan XU[1], Mr. Kevin BOWDEN[2], Ms. Jiaqi WU[2], Mr. Shan JIANG[1], Mr. Hanqing WU[1], Dr. Peiyuan ZHOU[1], Ms. Yanni YANG[1], Ms. Jiating ZHU[1] and Mr. Ruosong YANG[1] across my whole undergraduate research.

I want to thank my co-examiner Dr. Ajay Kumar PATHAK and 2nd assessor Dr. Richard LUI who spent their valuable time to assess this work. Last but not least, I would like to thank my beloved family and friends for their earnest love and support.

[1]Department of Computing, The Hong Kong Polytechnic University
[2]Natural Language and Dialogue Systems Lab, University of California Santa Cruz
[3]Geospatial Data Center, Massachusetts Institute of Technology
[4]Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the Internet and big data era, more and more data is collected from different Internet services. Users of social media and online information platform leave their reviews about products, movies, restaurants and so on to the Internet services providers like Amazon, IMDB, Yelp and Taobao. Based on the collected data, learning opinions included in the pure text of the users automatically becomes a hot demand. While, “sentiment analysis”, one aspect of “affective computing” becomes an urgent need in the business world (Yadollahi et al., 2017). Opinion analysis based on those reviews can not only help the business operators to understand the users and improve the services, but also provide reference information to the users to benefit their decision making.

Sentiment analysis refers to all methods of detecting, analyzing, and evaluating people’s state of mind towards events, issues, or any other interests (Yadollahi et al., 2017). Sentiment analysis algorithms with high accuracy have already been integrated into many Internet services and are applied in varied fields in people’s daily lives. For example, gaining information about customer satisfaction by scanning their reviews on Yelp[1] can help the platform improve search accuracy and enhance service qualities of restaurants (Feldman, 2013). Analyzing patients’ words can help

[1]Yelp is an online platform which publishes crowd-sourced reviews about local restaurants. Their website is: https://www.yelp.com/.

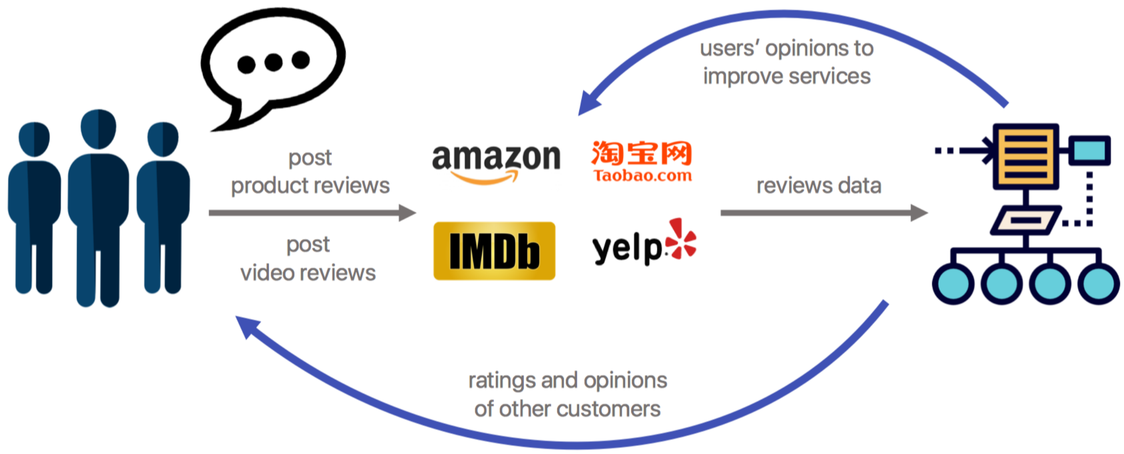


Figure 1: Review processing flow

to predict mental-health disorders (Yadollahi et al., 2017). Educational technology equipped with sentiment analysis techniques can select proper high-quality teaching materials by analyzing students' feedback (Pang et al., 2008).

> **Example: Sentiment analysis input and output (Yadollahi et al., 2017)**
>
> Input a review about a video: *"I find this short video is very interesting and funny! I love the main character very much!"*
> Output the sentiment of this review: positive / five scores out of five.

Among all the sentiment analysis areas, those focus on specific users and products have been discussed by researchers for a long time because of its potential application to the personalized recommendation systems (Chen et al., 2016). The reviews are normally written by a user to express his/her opinion on a product. The product can be goods, a restaurant or a movie and so on. Therefore, user preferences and product information would be crucial background information for the review documents.

There is rapid progress in sentiment analysis after the emergence of the deep neural network and related machine learning techniques. A lot of new methods based on neural network are proposed to increase the performance of prediction towards reviewers' sentiments (Wang and Manning, 2012).

In this project, we target at sentiment classification task by reflecting user profile and product information in the machine learning model. This model can not only reflect user or product features from different perspectives, but also keep the long-term features learned.

The rest of this report will be organized as follows. In the left part of this chapter, drawbacks of existing approaches, our final goal and the contribution of this project will be introduced. Chapter 2 states details of related knowledge and methods. Chapter 3 will introduce some fundamental models and techniques used in the new proposed model. Next, in the Chapter 4, the new proposed *Joint User and Product Memory Network (JUPMN)* will be introduced with detailed model architecture. Then, Chapter 5 shows experimental results and analysis. Finally, directions for future work and the conclusion is made in Chapter 6.

1.1 Problem Statement

In the field of sentiment classification, higher prediction result is always the target as higher accuracy can lead to larger potential commercial market. However, to achieve this goal, there are a few challenges both from the technical structure and real-world industrial needs in existing works:

1. **Using background information in sentiment analysis**
2. **Reflecting user profile and product information in different perspectives**
3. **Keeping long-term memory for document training**

1.1.1 Using Background Information in Sentiment Analysis

For the first problem, *background information* can be considered as associated information related to the main documents.

> **Example: What is background information**
>
> For sentiment analysis of posts on Twitter, users' profile, users' history behaviors and comments under posts written by other users can be considered as background information for the original posts.

Considering related background information for different sentiment analysis cases is important. Restaurants reviews, movie reviews, services reviews have different domain knowledge, and the reviews have different properties accordingly. Without considering different entities in sentiment classification task, it is not sufficient to reflect the real-world situation and make prediction (Gui et al., 2016). What's more, background information can provide more facts and possibilities for analyzing original documents which may improve the final results of sentiment classification.

In this project, for analyzing sentiment for reviews, background information contains two parts. The first part is *user profile*, which can be retrieved from past reviews posted by particular users; the second part is *product information*, which can be obtained by processing existing reviews about particular products.

1.1.2 Reflecting User Profile and Product Information in Different Perspectives

For the second problem, the background information may consist of different elements. Varied background elements may have separate effects on the reviews, and their influence varies. It is not proper to consider all these background elements in context as one united item.

> **Example: Different background information influence the results**
>
> In a review about video v posted by user u, u said "The movie is so good and touching". From the perspective of this user, u maybe has a mean personality, even the review content is somehow positive, but u only give two stars out of five. If the user u is lenient, then he/she maybe gives all the movies five stars. From the perspective of this video, the topic of v may be easy to touch people and make people emotional, even most of the reviews about v is very positive, but maybe the actual quality is only two stars out of five.

User's stand towards a product is very dependent on the preference of the specific user, and the stand is very subjective. On the other hand, product information may contain more objective information about the entity being reviewed. It is not appropriate to consider user profile and product information as a unified data entry, their influences towards the review documents shall be treated separately in the design of machine learning model.

1.1.3 Keeping Long-term Memory for Document Training

For machine learning problem related to text processing, the common problem is that it is hard for the model to keep the features from previously trained documents. In that case, a significant amount of document data will not benefit the learning, but it may harm the learning process. Long Short-Term Memory (LSTM) network are put forward by the research community to address this issue by utilizing a hidden state connecting through all cells in the network to keep the long-term features (Hochreiter and Schmidhuber, 1997). However, there is a still a space to improve the long-term memory performance to benefit the machine learning for documents.

1.2 Objective

According to the analysis of the problems above, the objective of this project is:

Proposing and evaluating a new sentiment analysis model. This model should utilize user profile and product information separately and preserve long-term features.

This model should be evaluated by commonly-used datasets among the research community and compare with other state-of-the-art models. Detailed analysis of the experimental results shall be accomplished to investigate the characteristics of the model and help to point out future directions.

1.3 Contribution

The main contributions of this work includes:

1. **Proposed new Joint User and Product Memory Network (JUPMN) model to learn user profiles and product information separately with joint prediction mechanism**
2. **Improved sentiment classification accuracy by 0.6%, 1.2% and 0.9% on IMDB, Yelp13 and Yelp14 datasets compared to the state-of-the-art method**
3. **Analyzed the effect of memory size, number of computational layers, weight of user and product memory network in the JUPMN**

The major part of this project is written as the following paper.

> Yunfei Long*, Mingyu Ma*, Rong Xiang, Qin Lu, Chu-Ren Huang. Fusing User Memory and Product Memory for Sentiment Classification. (*: Equal contribution)

Chapter 2

Related Works

Related works about this project can be mainly divided into four parts. Firstly, *Machine-learning-based sentiment analysis* in Section 2.1 is about methods for sentiment classification with deep neural networks; secondly, *Attention-based sentiment analysis models* in Section 2.2 is about the machine learning models utilizing related information as attention; thirdly, usage of *user and product information in sentiment analysis* in Section 2.3 is a thinking that combines user and product data with sentiment classification models; finally, *memory networks* in Section 2.4 is a new machine learning structure which fits the needs of long-term memory of text.

2.1 Machine-learning-based Sentiment Analysis

There are three levels of automatic classification of polarity categorized by granularity: document, sentence and aspect-level (Yadollahi et al., 2017). One method for sentiment analysis is to treat it as a special application of text classification. Traditional approaches to sentiment analysis use linear models or kernel methods on sentences representation based on sparse lexical features such as n-grams (Wang and Manning, 2012).

Nowadays, neural-network-based approaches targeted at sentiment classification are quite effective (Kim, 2014; Tang et al., 2014). Those methods provide end-to-end trainable models to learn the features from the pure text. Commonly-used machine learning models including Convolutional Neural Network (CNN) (Kim, 2014), Recursive auto-encoders (Socher et al., 2013) and Long Short-Term Memory(LSTM) network (Hochreiter and Schmidhuber, 1997; Tang et al., 2015a).

Especially, LSTM provides an end-to-end hidden state to learn the overall features, so it performs better to keep the previously trained features. Thus, LSTM model is very suitable for language processing task since the text has a sequential structure. Machine-learning-based models are widely used for many tasks like word embedding, sentiment analysis and dialog generation. For example, Tang et al. proposed a neural network model to learn sentiment-specific word embedding from data of Twitter (Tang et al., 2014).

2.2 Attention-based Sentiment Analysis Models

Normal machine learning models for Natural Language Processing (NLP) treat all the words equally from the start of the training process. While actually, some words are more important than others, and some words may provide additional sentiment information. To utilize more features and get better prediction result, some researchers proposed some attention-based models which can incorporate document structure information in the machine learning model design (Yang et al., 2016).

In the works of Long et al. and Mishra et al., they proposed an attention-based model utilizing cognitive eye-tracking data for sentiment analysis (Long et al., 2017; Mishra et al., 2017). These attention-based methods can lead the models to focus more on the words that contribute the most in defining sentiment of sentences.

2.3 User Profile and Product Information in Sentiment Analysis

Some researchers already combine additional user profile and product information in the machine learning model to achieve better performance for sentiment prediction. User personal preferences can affect their sentiment rating of a review, while product properties can also affect the result. It is vital to consider the user and product variance in the model design.

Tang et al. proposed User Product Neural Network (UPNN) utilizing user-sentiment and product-sentiment, and their performance proves that continuous user and product representations can increase accuracy for sentiment classification a lot (Tang et al., 2015b).

The work of Chen et al. (2016) shows that neural network that taking account of the global user preferences and product properties in both word level and sentence level can outperform state-of-the-art methods. The model proposed by Gui et al. (2016) learns the embedding of users and benefits the sentiment classification.

2.4 Memory Networks

Recently, some state-of-the-art works utilize "memory networks" to construct an end-to-end learning model so that interaction between different elements can be enabled (Weston et al., 2014; Sukhbaatar et al., 2015). The memory network is based on a recurrent neural network, where the recurrent reads from an external memory so that the model can better reflect features under a different context. Many applications of memory networks are introduced then.

A question answering solution is demonstrated by Sukhbaatar et al. (2015). The related sentences containing background knowledge of this question is considered as memory and are stacked in the external memory. Then the model can compare input question and background sentences to provide a reasonable answer to the question.

Tang et al. (2016) proposed a sentiment classification model utilizing deep memory networks with attention mechanism and explicit memory. Another deep memory networks model is introduced for attitude identification in which memory for target identification and memory for polarity classification enable different targets to interact in both target detection task and the polarity classification task (Li et al., 2017). Most related to this work, Dou proposed a deep memory network utilizing user and product information for sentiment classification (Dou, 2017). In Dou's model, user and product information compose the memory part to reflect the context in final rating prediction.

Chapter 3

Methodology

A few essential techniques and models will be utilized and referenced in this work. One is the way to obtain document representation shown in Section 3.1. Another methodology is the basic structure of the Memory Network shown in Section 3.2.

3.1 Hierarchical LSTM Model for Document Representation

In our model, the first step is to convert all documents to vectors which can be computed by the neural network. Especially, because this is a document-level representation task, it can not be solved by simple word embedding or bag-of-word method. Thus, we reference the solution proposed by Chen et al. (Chen et al., 2016) which can be considered as a proper approach for document representation. Figure 2 shows the model structure.

This model consists of two LSTM layers which convert the representation from word to sentence level and from sentence to document level. Assume $s\#$ is sentence sequence, $w\#$ is word sequence and $cell\#$ is LSTM cell number, then each single word is represented by a simple word vector shown as $w_{w\#}^{s\#}$ in the figure, and $h_{cell\#}^{s\#}$ in the figure is the memory vector passed among LSTM cells. The evaluation result shows

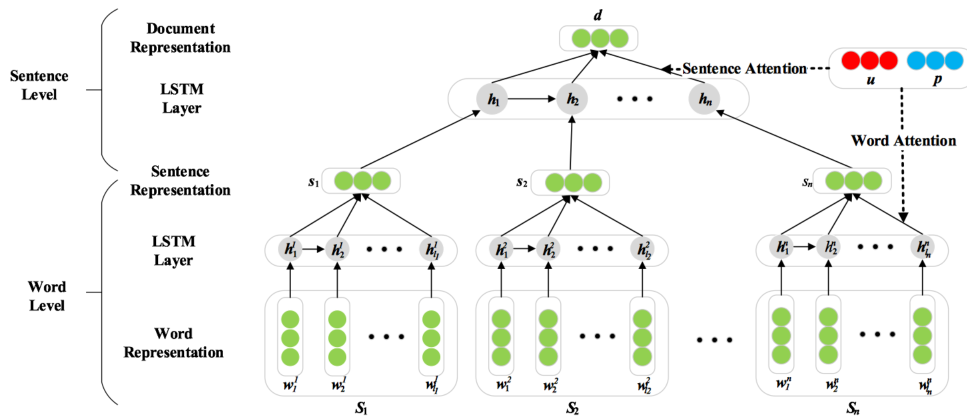


Figure 2: Structure of hierarchical LSTM network (Chen et al., 2016)

that this hierarchical LSTM model for document representation can achieve good result. The source code for this implementation in Python can also be used [1].

3.2 Basic Memory Network Model

After getting document representation, memory network will be the main container for local documents and background information to interact. An end-to-end memory network structure was proposed by Sukhbaatar et al. (2015), which shown in Figure 3. The (a) part shows the detailed structure in one hop, part (b) shows multiple hops structure. An implementation using Python and TensorFlow can be used [2].

There are four main components I, G, O and R in a memory network (Weston et al., 2014), for this specific memory network shown in Figure 3, they are:

1. I: input feature map which converts the incoming input to the internal feature representation. Embedding matrix B here is the feature map.

2. G: generalization which updates old memories given the new input. Embedding matrix A and C are generalizations here.

[1]https://github.com/thunlp/NSC
[2]https://github.com/carpedm20/MemN2N-tensorflow

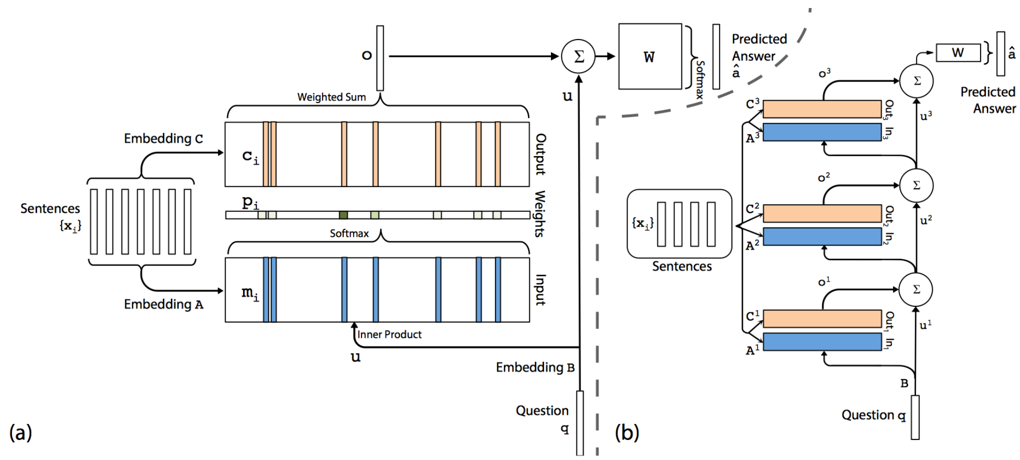


Figure 3: Structure of basic memory network (Sukhbaatar et al., 2015)

3. O: output feature map which produces a new output. o here is the output feature map, which outputs the memory representation.

4. R: the response which converts the output into the response format desired. Final softmax with weight matrix W is the response here.

Based on this basic structure of end-to-end memory network, an aspect-level sentiment analysis model utilizing memory network is chosen as the baseline work (Tang et al., 2016). Executable code is provided by Ganesh [3].

In Figure 4, we can have an insight of this attention-based model. This model has multiple computational layers (hops), each of which includes an attention layer and a linear layer. Each attention layer can select important elements from external memory according to the input aspect vector. After many hops, a prediction can be got through a *softmax* function.

[3] https://github.com/ganeshjawahar/mem_absa

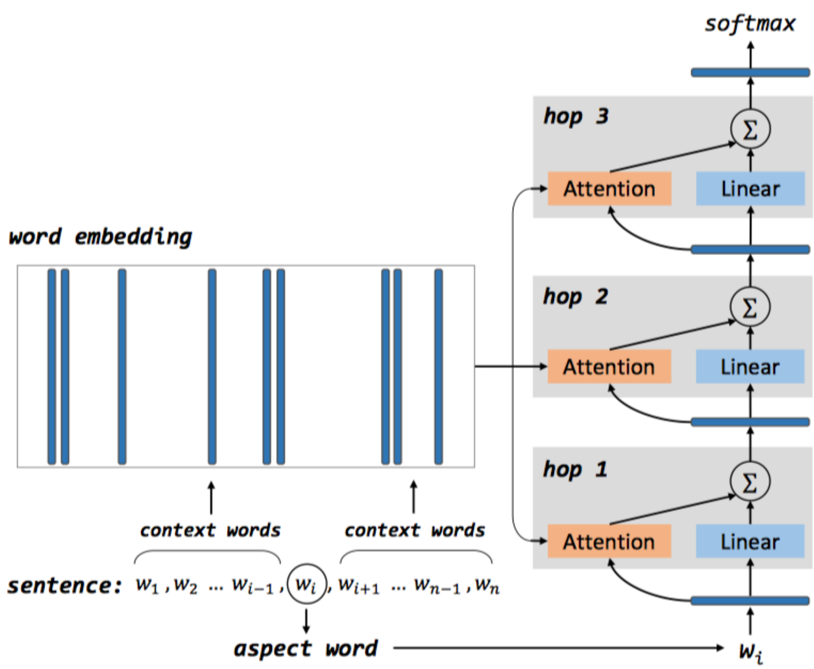


Figure 4: Structure of aspect-level sentiment analysis model (Tang et al., 2016)

Chapter 4

Model Design: JUPMN

To target at the problems in existing systems, we propose a new model for sentiment analysis called *Joint User and Product Memory Network (JUPMN)* to utilize user profile and product information in a better manner. Two essential parts form JUPMN. Firstly, a hierarchical LSTM network is constructed to obtain document representations for all review documents. Next, two memory networks are constructed for user profile and product information respectively, and then the cost function of two memory networks are combined. The final prediction can be output through a $softmax$ layer.

4.1 Task Definitions and Symbols

Basic symbols and definitions in this model will be defined in this section. Let D be all the review documents in the dataset, then U and P are all users and all products in this dataset. For each document d in D ($d \in D$), there is a user u ($u \in U$) who posted it, and this review d is talking about a product p ($p \in P$). So the u is the writer of d, and p is the target of d. We use $U(d)$ to represent all documents posted by u and $P(d)$ to represent all documents targeted at p. Therefore, $U(d)$ and $P(d)$ are background information about user profile and product information in term of d.

Our problem definition is:

> **Suppose a user u writes a document review d about product p, the discrete prediction score y for the review document d based on the input $< d, U(d), V(d) >$ should be output.**

4.2 Part 1: Document Embedding

The first half component of JUPMN is embedding all documents into numeric vectors. The document vectors can reflect semantic meanings in the documents and benefit the efficiency of model training in the left part of JUPMN.

One reliable model for document representation is using a hierarchical Long Short-Term Memory (LSTM) network. Hierarchical LSTM network is the state-of-the-model for document embedding (Chen et al., 2016). So it can reflect more semantic meaning in the output document representation compared to other models.

The input is all d, and the output is $\vec{d}$. With this process, we can also obtain vectorized background information documents vector stack $\hat{U}(d)$ and $\hat{P}(d)$.

Inspired by Chen et al. (2016), a two-layer LSTM network is employed. These two layers are corresponding to conversion from word-level to sentence-level and conversion from sentence-level to document-level. An LSTM layer is used to first obtain sentence representations by picking up the hidden state of the first LSTM layer. The same mechanism is used to obtain document-level representations from sentence-level representations in this document.

Assume the embedding dimension is n, then each word is embedded into a vector with size $1 \times n$. Assume a sentence have num_w words inside, then the input matrix to the first LSTM layer is of size $num_w \times n$. From the hidden state of the first LSTM layer, a vector of this sentence with size $1 \times n$ can be obtained which is considered as the sentence representation. Assume a document contains num_s sentences in it, and each sentence is embedded into a $1 \times n$ vector, then the input to the second LSTM layer would be a $num_s \times n$ matrix. From the hidden state of the second LSTM layer, we can obtain the vector with size $1 \times n$ which is considered as the final representation of the document.

The hidden state of these two LSTM layers will be optimized by learning more documents with more training iterations.

Based on the hierarchical LSTM network, attention can be added to improve the document representations further. The user or product attention can be added following the work of Chen et al. (2016). Inspired by Long et al. (2017), cognition data can be added as extra attention on the review documents. In summary, the following three models can be used to obtain document embeddings using hierarchical LSTM, the second method JUPMN [LSTM+UPA] is the default embedding approach.

1. JUPMN [LSTM]: document representations are obtained by hierarchical LSTM network without any attention
2. JUPMN [LSTM+UPA]: documents representations are generated by hierarchical LSTM with user and product attention
3. JUPMN [LSTM+UPA+CBA]: documents representations are generated by hierarchical LSTM with eye-tracking data attention as well as user and product attention

4.3 Part 2: Memory Network Structure

Figure 5 shows the memory network architecture of the JUPMN model. Memory network part of JUPMN consists of two memory networks: *User Memory Network (UMN)* and *Product Memory Network (PMN)*. Each memory network connects to external memory documents $\hat{U}(d)$ or $\hat{P}(d)$ respectively and then be combined together

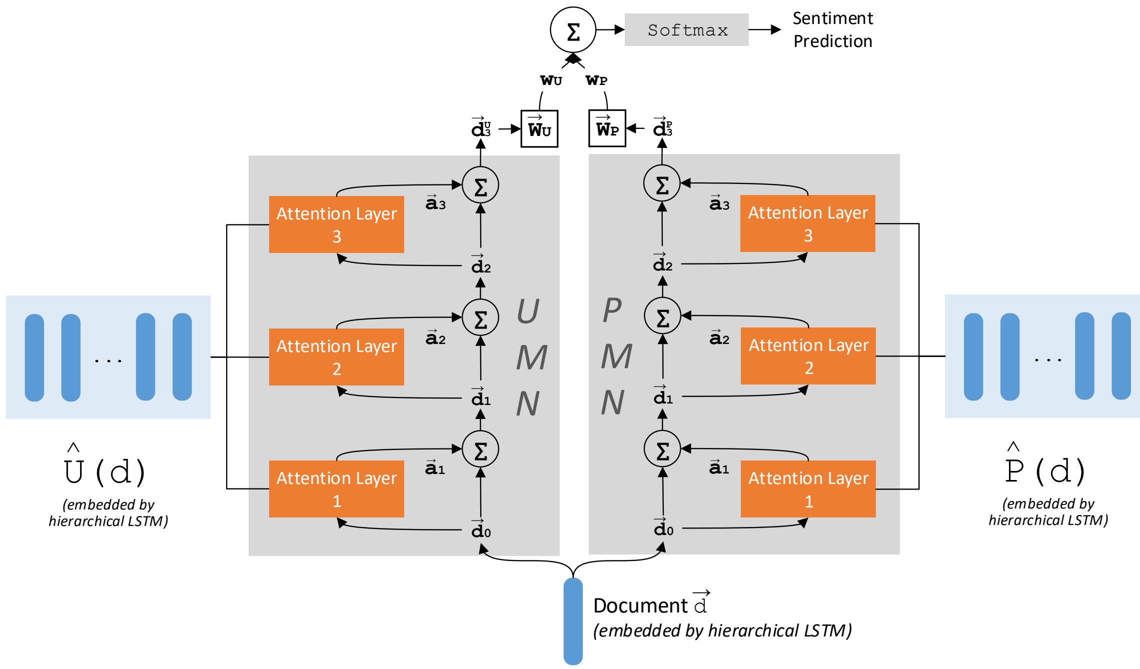


Figure 5: Model Structure of JUPMN

finally.

For each external memory, a list of document vectors is stacked to simulate the context of the user profile or product information. Assume m_{actual} is the number of documents posted by user u or targeted at product p, then the external memories $\hat{U}(d)$ and $\hat{P}(d)$ are in size $n \times m_{actual}$ where n is the dimension size. Maximum memory size is fixed marked as m, so if the $m_{actual} \geq m$, the first m documents will be used for memory construction and the size of external memory is $n \times m$.

Each memory network consists of K computational layers (hops), and an attention layer and a linear layer form each hop. The input document vector $\vec{d}$ will be fed into two memory networks as the input to the first hop($\vec{d_0} = \vec{d}$). Among all the K hops, for kth hop, each $\vec{d_{k-1}}$ go through an attention layer which will be introduced later to get the output of the attention layer $\vec{a_k}$. Then $\vec{a_k}$ is linearly added to d_{k-1} to become the output of this hop as d_k and input to next hop for further computation.

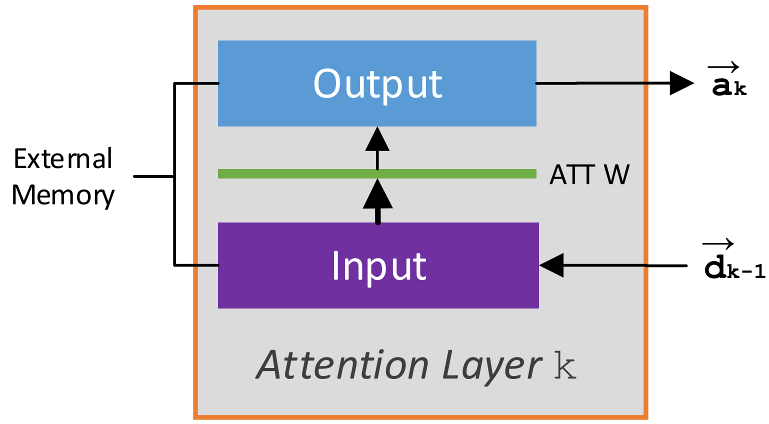


Figure 6: Attention layer in JUPMN

The detailed structure of the attention layer is shown in Figure 6. The input memory and output memory shown in Figure 6 is exactly the external memory which is noted as $\hat{M}$. The comparison result between the input document $\vec{d_{k-1}}$ and the document vectors in the external memory $\hat{M}$ will be calculated as a attention weight vector $\vec{p}_k$ according to Equation 1.

$$\vec{p}_k = Softmax(\vec{d}_{k-1}^T * \hat{M}) \tag{1}$$

Then the output of the attention layer $\vec{a}_k$ is the output memory documents weighted by the attention weight $\vec{p}_k$ with a linear addition. Equation 2 shows the approach to calculate output of the attention layer from the attention weight and output memory documents.

$$\vec{a}_k = \sum_{i=0}^{m} p_{k_i} * \vec{M}_i. \tag{2}$$

After going through the Kth hop, the output of UMN $\vec{d}_K^u$ and PMN $\vec{d}_K^p$ is combined together by a combination mechanism. There are two possible combination mechanisms, one is adding weighted output to produce the output of JUPMN following

Equation 3.

$$Output_{JUPMN} = \vec{W}_U \vec{d}_K^u + \vec{W}_P \vec{d}_K^p \tag{3}$$

Another possible combination mechanism is adding weighted output with constants w_U or w_P to produce the final output as shown in Equation 4. The constants w_U or w_P can reflect the importance of UMN or PMN in this review document dataset, and they can be optimized during the training. By default, the second combination mechanism is used.

$$Output_{JUPMN} = w_U \vec{W}_U \vec{d}_K^u + w_P \vec{W}_P \vec{d}_K^p \tag{4}$$

Finally, the sentiment prediction can be obtained by a $Softmax$ function shown in Equation 5.

$$Sentiment\ Prediction = Softmax(Output_{JUPMN}) \tag{5}$$

The parameters in the model include $\vec{W}_U$, $\vec{W}_P$, w_U and w_P. By minimizing the loss function between the sentiment predictions and ground truth sentiment labels, those parameters can be optimized.

4.4 Variations of JUPMN

Besides the original JUPMN structure, there are two variations with different configurations based on basic JUPMN and they are noted as following symbols.

- JUPMN-U: the JUPMN model with only User Memory Network, disable Product Memory Network
- JUPMN-P: the JUPMN model with only Product Memory Network, disable User Memory Network

Chapter 5

Evaluation and Analysis

The model design shown in Chapter 4 is implemented with `Python` and `TensorFlow`. The source code is available at https://github.com/derekmma/jupmn.

The datasets for testing and performance metrics are introduced in Section 5.1. Then two types of evaluation are performed. The first type is to compare JUPMN with other existing systems shown in Section 5.2. In Section 5.3, second type of evaluation is to compare different variations of JUPMN in terms of the following four aspects:

- Number of computational layers (Section 5.3.2);
- Memory size (Section 5.3.3);
- Combination mechanism of user/product memory network (Section 5.3.4) and
- Influences of user profile and product information (Section 5.3.5).

Finally, a case study to show the characteristics of JUPMN is demonstrated in Section 5.4.

5.1 Benchmark Datasets and Performance Metrics

In the evaluation and verification part, we use three datasets that are already tested by other existing models to compare with JUPMN. IMDB dataset which is derived from IMDB by Diao et al. (2014) and Yelp datasets which are from Yelp Dataset Challenge in 2013 and 2014 [1] and cleared up by Tang et al. (2015a) will be used. These three datasets have been used in many works about sentiment classification such as Dou (2017); Tang et al. (2015b); Long et al. (2017); Chen et al. (2016).

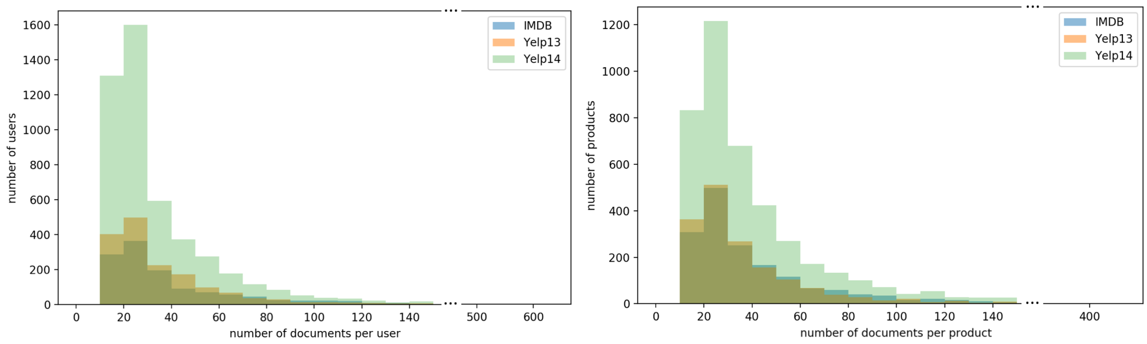
A statistic of three benchmark datasets are shown in the Table 1. The distribution of document numbers per user or product is shown in Figure 7. We can find that the distribution of data follows the long-tail distribution, most of the documents' numbers are within 1-100 per user or product.

	IMDB	Yelp13	Yelp14
number of classes	10	5	5
number of review documents	84,919	78,966	231,163
number of users	1,310	1,631	4,818
number of products	1,635	1,631	4,194
average sentences' length	24.56	17.37	17.25
average number of documents per user	64.82	48.41	47.97
average number of documents per product	51.93	48.41	55.12
number of users with 0-50 reviews	1,037	1,302	3,875
number of users with 50-100 reviews	126	249	711
number of users with 100-150 reviews	69	52	129
number of users with 150-200 reviews	30	17	52
number of products with 0-50 reviews	1,223	1,299	3,150
number of products with 50-100 reviews	318	254	749
number of products with 100-150 reviews	72	56	175
number of products with 150-200 reviews	22	24	120

Table 1: Statistics of benchmark datasets

Then the performance metrics that we used to measure the performance of the models are defined as following. $Accuracy$, MAE and $RMSE$ will be the measures for divergences between results of different models. They are defined as following equations 6, 7 and 8. T is number of correct predictions, N is the size of the testing set, and

[1] http://www.yelp.com/dataset_challenge



(a) Statistic of documents # per user (b) Statistic of documents # per product

Figure 7: Number of documents per user/product for three datasets

py_i and gy_i are prediction and ground truth for each training or testing document.

$$Accuracy = \frac{T}{N} \tag{6}$$

$$MAE = \frac{\sum_i |py_i - gy_i|}{N} \tag{7}$$

$$RMSE = \sqrt{\frac{\sum_i (py_i - gy_i)^2}{N}} \tag{8}$$

5.2 JUPMN and Comparison Models

In this section, the best experimental result of JUPMN will be compared with comparison models to analyze the overall ability of JUPMN.

5.2.1 Comparison Models

Most commonly used models can be categorized into three groups. The first group methods are simple methods based on language features (Chen et al., 2016). They are:

- **Majority** — A simple majority classifier based on sentence labels;

- **Trigram** — A SVM classifier using unigram/bigram/trigram as features;
- **Text feature** — A SVM classifier using word level and context level features, such as n-gram and sentiment lexicons;
- **AvgWordvec** — A SVM classifier that takes the average of word embeddings in Word2Vec as document embedding.

The second group of methods are those with machine learning techniques, including:

- **SSWE** (Tang et al., 2014) — A SVM model using sentiment specific word embedding;
- **RNTN+RNN** (Socher et al., 2013) — A Recursive Neural Tensor Network (RNTN) to represent sentences and trained using RNN;
- **CLSTM** (Xu et al., 2016) — A Cached LSTM model to capture overall semantic information in long text;
- **LSTM+LA** (Chen et al., 2016) — A state-of-the-art LSTM model using local context as attention mechanism in both sentence-level and document-level;
- **LSTM+CBA** (Long et al., 2017)— A LSTM model using cognition based data to build attention mechanism.

The third group includes the state-of-the-art models using both user profile and product information, including:

- **UPNN** (Tang et al., 2015b) — User and product information for sentiment classification at document level based on CNN network;
- **UPDMN(K)** (Dou, 2017) — A deep memory network for document-level sentiment classification which captures user and product information by a unified model;
- **InterSub** (Gui et al., 2016) — A CNN model making use of network embedding of user and products;

- **LSTM+UPA** (Chen et al., 2016) — The state-of-the-art LSTM model including both local context based attentions and user/product in the attention mechanism at both sentence-level and document-level.

5.2.2 Experimental Results

The experimental results of JUPMN and the performance of comparison models declared in corresponding papers are listed in Table 2. For the JUPMN performance in this table, memory size is set at 100, and there is one hop in the memory networks.

	IMDB			Yelp13			Yelp14		
Model	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
Majority	0.196	2.495	1.838	0.392	1.097	0.779	0.411	1.06	0.744
Trigram	0.399	1.783	1.147	0.577	0.804	0.487	0.569	0.814	0.513
TextFeature	<u>0.402</u>	<u>1.793</u>	<u>1.134</u>	<u>0.572</u>	<u>0.800</u>	<u>0.490</u>	<u>0.556</u>	<u>0.845</u>	<u>0.520</u>
AvgWordvec	0.304	1.985	1.361	0.530	0.893	0.562	0.526	0.898	0.568
SSWE	0.312	1.973	N/A	0.549	0.849	N/A	0.557	0.851	N/A
RNTN+RNN	0.400	1.734	N/A	0.574	0.804	N/A	0.582	0.821	N/A
CLSTM	0.421	1.549	N/A	0.592	0.729	N/A	0.637	0.686	N/A
LSTM+LA	0.443	1.465	N/A	0.627	0.701	N/A	0.637	0.686	N/A
LSTM+CBA	<u>0.489</u>	<u>1.365</u>	N/A	<u>0.638</u>	<u>0.697</u>	N/A	<u>0.641</u>	<u>0.678</u>	N/A
UPNN(K)	0.435	1.602	0.979	0.608	0.764	0.447	0.596	0.784	0.464
UPDMN(K)	0.465	1.351	0.853	0.613	0.720	0.425	0.639	0.662	0.369
InterSub	0.476	1.392	N/A	0.623	0.714	N/A	0.635	0.690	N/A
LSTM+UPA	<u>0.533</u>	**1.281**	N/A	<u>0.650</u>	<u>0.692</u>	N/A	<u>0.667</u>	<u>0.654</u>	N/A
JUPMN	**0.539**	<u>1.283</u>	**0.725**	**0.662**	**0.667**	**0.375**	**0.676**	**0.641**	**0.351**

Table 2: Experimental results of JUPMN and comparison models[2]

5.2.3 Findings

The first and most important finding from the Table 2 is that JUPMN outperforms all the existing systems, even the state-of-the-art LSTM+UPA model. Regarding IMDB dataset, JUPMN achieves better accuracy than the state-of-the-art model by 0.6%, the accuracy increment than the state-of-the-art model for Yelp13 and Yelp14

[2]Best results are marked in bold; second best are underlined in the table

datasets are 1.2% and 0.9% respectively.

The second finding is that the models in group 2 perform better than group 1 generally, and the models in group 3 outperform group 2 generally. From the comparison between group 1 and group2 we can find, advanced machine learning methods are more effective than traditional SVM classifier with language feature engineering. From the comparison between group 2 and group 3, we can infer that user and product information is beneficial for improvement of sentiment classification.

There are a few exceptions. Some feature engineering models can achieve comparable performance with some machine learning methods. For example, "TextFeature" model beats "SSWE". Some deep learning method without user and product information can still achieve good result compared to the ones with user and product information. An instance is "LSTM+CBA" (Long et al., 2017) utilizes local cognition based data as attention which outperforms models with the user and product information like "UPNN", "UPDMN" and "InterSub".

5.3 JUPMN with Different Configurations

To feature out the best configuration and find out insights in the JUPMN model, three sets of experiments are employed to test the effects of the following configurations:

1. Number of computational layers (hops)
2. Memory size (number of documents vectors in external memories)
3. Weighting mechanism used when combining user/product memory networks
4. Importance of User/Product Memory Network

5.3.1 Experimental Results

First set of experiments tests the JUPMN with different configurations including the variations with only User Memory Network (UMN) or Product Memory Network

(PMN) in Figure 5 and the models with the different number of computational layers (hops) in the memory networks. The experimental results are shown in the Table 3. This set of experiments are conducted when memory size is 100, and by default use document representations with user and product attention.

	IMDB			Yelp13			Yelp14		
	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
JUPMN-U(1)	<u>0.536</u>	**1.283**	<u>0.737</u>	0.656	0.687	0.380	0.667	0.655	0.361
JUPMN-U(2)	0.526	<u>1.285</u>	0.748	0.653	0.689	0.382	0.665	0.661	0.369
JUPMN-U(3)	0.524	1.295	0.754	0.651	0.692	0.388	0.661	0.667	0.374
JUPMN-P(1)	0.523	1.346	0.769	<u>0.660</u>	<u>0.668</u>	**0.370**	<u>0.670</u>	<u>0.649</u>	<u>0.357</u>
JUPMN-P(2)	0.517	1.348	0.775	0.656	0.680	0.380	0.667	0.656	0.364
JUPMN-P(3)	0.512	1.356	0.661	0.651	0.699	0.388	0.661	0.661	0.370
JUPMN(1)	**0.539**	**1.283**	**0.725**	**0.662**	**0.667**	<u>0.375</u>	**0.676**	**0.641**	**0.351**
JUPMN(2)	0.522	1.299	0.758	0.650	0.700	0.390	0.667	0.650	0.359
JUPMN(3)	0.502	1.431	0.830	0.653	0.686	0.382	0.658	0.668	0.371

Table 3: Experimental results of JUPMN with different memory network hops and user and product information utilization[3]

Second set of experiments shows the performance of JUPMN under different memory sizes. The result of the experiments with different memory size all with only 1 hop shows in the Table 4.

Memory	IMDB			Yelp13			Yelp14		
Size	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
10	0.501	1.572	0.892	0.625	0.788	0.467	0.647	0.692	0.397
20	0.503	1.550	0.866	0.631	0.778	0.456	0.651	0.684	0.384
30	0.516	1.383	0.791	0.643	0.707	0.397	0.668	0.661	0.362
40	0.524	1.367	0.778	0.647	0.695	0.390	0.674	0.641	0.351
50	0.528	1.368	0.769	0.654	0.680	0.379	0.671	0.653	0.356
75	0.529	1.339	0.768	0.655	0.690	0.384	0.674	0.653	0.354
100	0.539	1.283	0.725	0.662	0.667	0.375	0.676	0.641	0.351

Table 4: Experimental results of JUPMN with different memory size

[3]Best results are marked in bold; second best are underlined in the table

The third set of experiments shows the best performance of JUPMN model with different combining mechanisms. The “JUPMN(not weighted)” is the JUPMN model without weighted combining results of two memory networks. The combining equation for “JUPMN(not weighted)” is Equation 3. The “JUPMN” shown in following Table 5 is the default model with weighted mechanism following Equation 4. The experiment results are listed in Table 5.

	IMDB			Yelp13			Yelp14		
Model	Acc	RMSE	MAE	Acc	RMSE	MAE	Acc	RMSE	MAE
JUPMN(not weighted)	0.538	1.289	0.737	0.656	0.682	0.379	0.670	0.645	0.354
JUPMN	**0.539**	**1.283**	**0.725**	**0.662**	**0.667**	**0.375**	**0.676**	**0.641**	**0.351**

Table 5: Experimental results of weighted and unweighted JUPMN

5.3.2 Analysis on Number of Computational Layers

According to the test shown in Table 3 where shows the performance result when the number of computational layers is 1, 2 or 3 for JUPMN. It is clear that when there is only one hop, the performance is the best compared to the cases with more hops. In the meantime, the models with 2 hops perform better than the ones with 3 hops. This observation can be explained by possible over-fitting and data distortion caused by structured text data.

5.3.3 Analysis on Memory Size

According to the data shown in Table 4, the accuracy of JUPMN under different memory sizes are drawn in the Figure 8.

Some user or product may have only a few related documents, so if the memory size is small, then some of the user or product may have empty of limited-size external memory. From the Figure 8, when the memory size increases from 10 to 75, the performance increases linearly. This fact also shows that the external memory indeed helps for the sentiment classification. While after the memory size is larger than 75, the performance does not increase anymore. So we can find that the memory size of

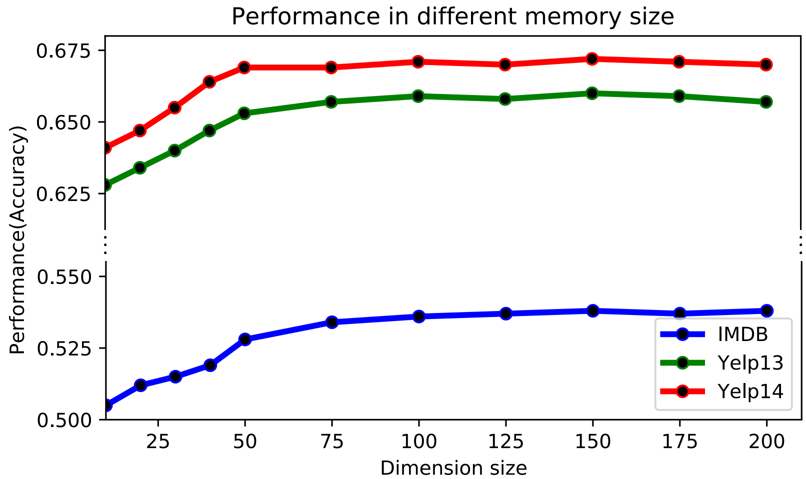


Figure 8: Accuracy of JUPMN under different memory size

100 is sufficient for these three benchmark datasets. This fact can also be explained since the average number of documents per user or product is around 75 according to the datasets' statistics shown in Table 1, so larger memory size will not help since there are not enough documents to fill in the memory space.

5.3.4 Analysis on Combining Weighting Mechanism

The finding from Table 5 is that the JUPMN variation which just adding results of two memory network together ("JUPMN(not weighted)") is not good as the JUPMN model with weights for combining (the default JUPMN model). From this finding, we can know that the weighted mechanism helps to balance the influences of the User Memory Network and Product Memory Network and can improve the performance of the model.

5.3.5 Analysis on Influences of User Profile and Product Information

From observing the experimental results of JUPMN-U and JUPMN-P shown in Table 3, we can find that for the restaurants' review datasets like Yelp13 and Yelp14, JUPMN-P works better than JUPMN-U. For Yelp13, JUPMN-P's performance is very close to the full model with two memory networks. While, for movie review dataset IMDB, JUPMN-U performs better than JUPMN-P. The initial conclusion from experimental results is that product information dominates more on the restaurants' reviews, user profile dominates more on the movies' reviews.

To verify this initial conclusion and investigate the influences of the user profile and product information, further analysis is employed by two approaches. One is checking the change of combining weights during the training process; the other one is plotting the word frequency categorized by user or product.

Investigation by Checking Combining Weights

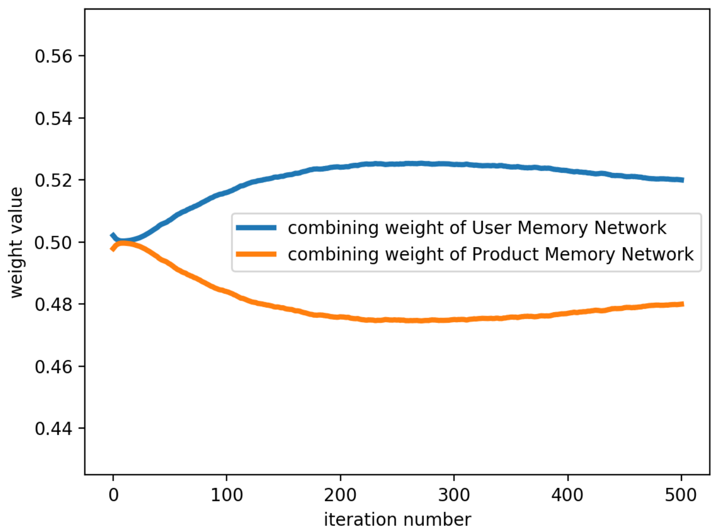
In the design of JUPMN, there are two weights when combining User Memory Network (UMN) and Product Memory Network (PMN) together as shown in Equation 4. w_U and w_P are two parameters used to weight the importance of two networks, so that their optimized value through the whole learning process can reflect the impacts of two memory networks to the final sentiment prediction.

The importance of UMN or PMN can be measured by the w'_U and w'_P which can be obtained by a simple mathematic transformation shown in the Equation 9 and 10.

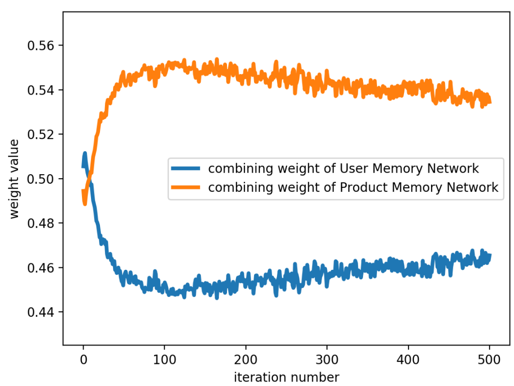
$$w'_U = \frac{w_U}{w_U + w_P} \tag{9}$$

$$w'_P = \frac{w_P}{w_U + w_P} \tag{10}$$

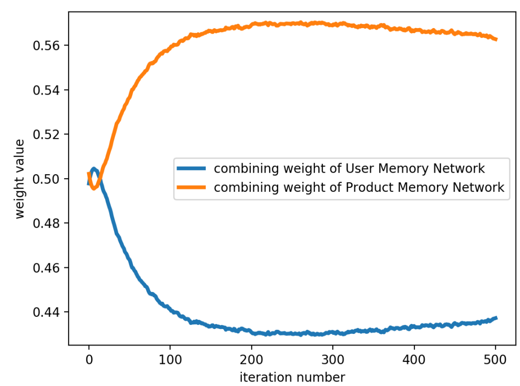
The Figure 9 shown the change of w'_U and w'_P in a learning process of JUPMN for all three datasets. The Table 6 shows the average combining weight w'_U and w'_P for all three benchmark datasets.



(a) for IMDB dataset



(b) for Yelp13 dataset



(c) for Yelp14 dataset

Figure 9: The change of w'_U and w'_P in a learning process of JUPMN for datasets

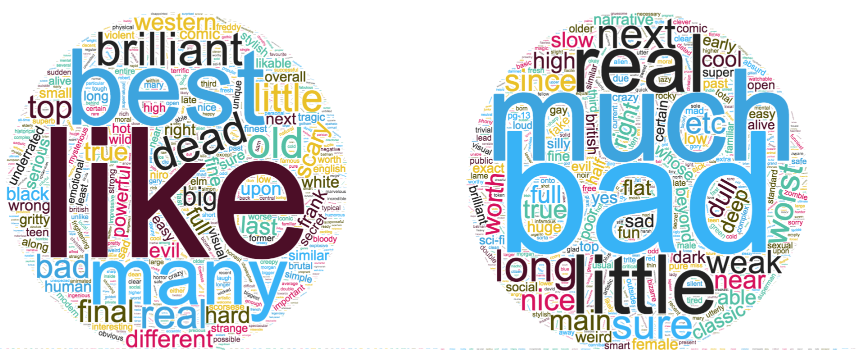
From the combing weight for all three datasets, the initial conclusion can be verified. For the Yelp13 and Yelp14, combining weight for PMN is larger than the combining weight for UMN, which verify that in restaurants review datasets, product information dominates more. While for the IMDB dataset, the combining weight for UMN is larger than the combining weight for PMN, which shows for movie reviews, user profile matters more.

IMDB		Yelp13		Yelp14	
w'_U	w'_P	w'_U	w'_P	w'_U	w'_P
0.534	0.466	0.475	**0.525**	0.436	**0.564**

Table 6: Average combining weight for benchmark datasets

Investigation by Word Frequency Plotting

To investigate the effect of the user profile and product information in the JUPMN, we can pick up some edge cases documents to analyze their properties. Firstly, to analyze the user and product information's influences on movie reviews, we pick all the reviews posted by 10 users who give average highest ratings among all users, and plot their word frequency in Figure 10a [4]. We also pick the 10 users who give average lowest ratings, and plot the word frequency in their reviews in Figure 10b.



(a) 10 users who give average highest ratings

(b) 10 users who give average lowest ratings

Figure 10: Word frequency for reviews of extreme users for IMDB dataset

From the plot, we can find that it's very easy for us to distinguish these two groups of users, the most generous users and the most rigorous users. The best raters use lots of positive words like *"like"*, *"best"*, *"brilliant"* and so on. While the worst raters use lots of negative works like *"bad"*.

We also plot the documents of extreme products in Figure 11. The word frequencies

[4]The word frequency diagrams are generated by https://www.wordclouds.com/

for the reviews of the best/worst movies, the movies with highest/lowest ratings generally, are shown in Figure 11a and 11b respectively.

(a) 10 movies with average highest ratings

(b) 10 movies with average lowest ratings

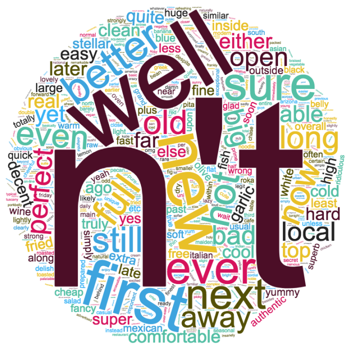
Figure 11: Word frequency for reviews of extreme products for IMDB dataset

According to Figure 11, the reviews for best and worst movies use a lot of objective words for movie description like *"first"*, *"old"*, *"new"* and so on. So it's hard to distinguish the best and worst movies from the work frequency diagram.

From the observation of Figure 10 and 11, we can find that user preference matters more for the movie reviews, and product information may not influence the result that much.

We apply the same analysis on movie review dataset to restaurants reviews. Figure 12 shows the word frequency for the review of the raters with highest or lowest rating, and the Figure 13 shows the word frequency of the reviews of the best or worst restaurants.

From the observation of word frequency of restaurants reviews filtered by user or product as shown in Figure 12 and 13, the word frequency for best or worst restaurants in terms of product is quite different. The common words used for the restaurants with high ratings are *"like"*, *"nice"*, *"fresh"* and so on, while the common words for worst ones are *"bad"* and so on. While for the users who give highest/lowest ratings, it's hard to distinguish them by word frequency since their commonly-used words are



(a) 10 users who give average highest ratings

(b) 10 users who give average lowest ratings

Figure 12: Word frequency for reviews of extreme users for Yelp13 dataset

(a) 10 restaurants with average highest ratings

(b) 10 restaurants with average lowest ratings

Figure 13: Word frequency for reviews of extreme products for Yelp13 dataset

quite similar. So we can conclude that for restaurants reviews, product information can reflect sentiment of users more compared to user profile.

From the word frequency analysis for movie reviews and restaurants reviews above, we can verify our initial conclusion proposed in the beginning of this section (5.3.5) as well.

5.4 Case Study

An example document is demonstrated in this section to show the characteristics of JUPMN of its difference compared to traditional LSTM. The following document is posted by a user on IMDB to comment on a science fiction movie.

> **Example: Example document**
>
> True sentiment label: 10 (most positive)
> Predicted sentiment by LSTM network: 1 (most negative)
> Predicted sentiment by JUPMN: 10 (most positive)
>
> Original review text:
> *okay, there are two types of movie lovers: ... they expect to see a Titanic every time they go to the cinema ... this movie sucks? ... it is definitely better than other sci-fi films ..... the audio and visual effects are simply terrific and Travolta's performance is brilliant-funny and interesting. what people expect from sci-fi movies is beyond me ... the rating for Battlefield Earth is below 2.5, which is unacceptable for a movie with such craftsmanship. Scary movie, possibly the worst movie of all time - including home made movies, has a 6! maybe we should all be a little more subtle when we criticize movies like this and especially sci-fi movies, since they have become an endangered genre ... give this movie the recognition it deserves.*

In this example document, the user tries to express his/her opinion in a complicated way. He/she firstly says a lot of negative words like "unacceptable", "worst", to cite other people's negative opinions on this movie. Then he makes a turn to express his love of this movie. Compared to ridiculous wrong prediction result given by traditional LSTM model, our JUPMN gives the correct sentiment prediction.

The possible explanation of this example is that JUPMN can learn the user preference from his/her past reviews towards other movies which are stacked in external memories. Through scanning other reviews, we can find out this user is actually a fan of science fiction movies. This feature may help the network to make the correct prediction. The Product Memory Network helps as well. JUPMN can compare this document with other reviews made about this movie to compare their similarity, so the features from product side may also contribute to the correct prediction result.

Chapter 6

Conclusion and Future Work

To better utilize user profile and product information in the sentiment classification task, a new model is proposed to address the issues such as deploying background information for sentiment classification and reflecting them in reasonable separated ways in the machine learning model.

In this project, a new machine learning model based on memory network called "Joint User and Product Memory Network (JUPMN)" is proposed for reflecting features of the user profile and product information from different perspectives in the machine learning model. In this model, a hierarchal LSTM network is firstly constructed to obtain document representations from the hidden states of LSTM networks, and then the document vectors are fed into the separated memory networks connected to external memory documents from users' perspective and products' perspective. The final combination mechanism will weight the results from two memory networks and output the final sentiment prediction.

Evaluation result shows that JUPMN outcomes all existing systems regarding the performance of sentiment classification task on three benchmark datasets IMDB, Yelp13 and Yelp14 and achieves significant improvements. Further analysis shows some interesting findings. Single hop structure leads to best performance, and user profiles influence more on movie reviews dataset while product information influences more on restaurants reviews by analysis of combining weight and word frequency. These analysis works also prove our hypothesis that user profile and product information influence the sentiment in different ways.

There are some possible directions for the future works. Firstly, aspect-level sentiment analysis tasks with memory network can be explored. Current work addresses the memory issue and background information issue from the document level, if these features can be embedded into aspect level, then further performance improvement may be possible.

Secondly, more knowledge can be combined into the memory network structure to further improve the performance. For example, eye-tracking data can be directed considered as part of the external memory and be used as separated memory network. In the current JUPMN structure, documents posted by the user and documents targeted at the product is considered as background context to form the external memories. However, advanced user profile like gender, location, history or advanced product information like origin, price and so on can also be used as features in the memory network.

Thirdly, the JUPMN can be expanded to be utilized in review document datasets in other languages like Mandarin and Cantonese.

Fourthly, memory network can not only apply to sentiment classification task, but also other Natural Language Processing tasks like dialog understanding and generation, emotion prediction and so forth.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. Neural sentiment classification with user and product attention. In *EMNLP*, pages 1650–1659, 2016.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM, 2014.

Zi-Yi Dou. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 532–537, 2017.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

Ronen Feldman. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89, 2013.

Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. Modeling user leniency and product popularity for sentiment classification. In *IJCNLP*, pages 1107–1111, 2013.

Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*, 2017.

Lin Gui, Ruifeng Xu, Yulan He, Qin Lu, and Zhongyu Wei. Intersubjectivity and sentiment: From language to knowledge. In *IJCAI*, pages 2789–2795, 2016.

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

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

Cheng Li, Xiaoxiao Guo, and Qiaozhu Mei. Deep memory networks for attitude identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 671–680. ACM, 2017.

Yunfei Long, Lu Qin, Rong Xiang, Minglei Li, and Chu-Ren Huang. A cognition based attention model for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 473–482, 2017.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. Leveraging cognitive features for sentiment analysis. *arXiv preprint arXiv:1701.05581*, 2017.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):694–707, 2016.

Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

Zhida Sun, Mingfei Sun, Nan Cao, and Xiaojuan Ma. Videoforest: interactive visual summarization of video streams based on danmu data. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, page 10. ACM, 2016.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.

Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015a.

Duyu Tang, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL (1)*, pages 1014–1023, 2015b.

Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*, 2016.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015c.

Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1722–1731, 2017.

Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics, 2012.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuangjing Huang. Cached long short-term memory neural networks for document-level sentiment classification. *arXiv preprint arXiv:1610.04989*, 2016.

Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R Zaiane. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, 50(2):25, 2017.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. Hierarchical attention networks for document classification. In *HLT-NAACL*, pages 1480–1489, 2016.