# Outline Extraction with Question-specific Memory Cells

JINGXUAN YANG*, Beijing University of Posts and Telecommunications, China
HAOTIAN CUI*, Tsinghua University, China
SI LI†, Beijing University of Posts and Telecommunications, China
SHENG GAO, Beijing University of Posts and Telecommunications, China
JUN GUO, Beijing University of Posts and Telecommunications, China
ZHENGDONG LU, DeeplyCurious.ai, China

Online consultation has been applied extensively in lots of professions, with the rapid development of Internet. Machine automatically extracts key information from individual cases, and provides individualized advice. In the important outline extraction process, semantic analysis is crucial to answer questions predefined by experts. We present a novel question-specific memory cell (QSMC) network to manipulate information on-the-fly as it reads texts. A particular question vector is integrated into each memory cell to indicate what information the cell focuses on. A sequentially updated state vector is also associated with each cell as question related sentence representation. We add a penalization term in loss function to make extracted knowledge more reasonable and interpretable. To support this study, we construct a new outline extraction corpus, **InjuryCase**, which is composed of 3995 real word Chinese occupational injury cases. Results show that our method makes a significant improvement. We further demonstrate the proposed framework on other two multi-aspect extraction tasks, and find that the proposed model also remarkably outperforms existing state-of-the-art methods. The source code and occupational injury corpus is publicly available[1].

CCS Concepts: • **Computing methodologies** → **Information extraction**; • **Information systems** → *Data mining*; • **Applied computing** → *Document management and text processing*;

Additional Key Words and Phrases: Outline Extraction, Memory Cell Network, Aspect Extraction

**ACM Reference Format:**
Jingxuan Yang*, Haotian Cui*, Si Li†, Sheng Gao, Jun Guo, and Zhengdong Lu. 2018. Outline Extraction with Question-specific Memory Cells. 1, 1 (July 2018), 17 pages. https://doi.org/0000001.0000001

* Equal contribution.
† Corresponding author.
[1] https://github.com/NingNingYang/qsmc.

Authors' addresses: Jingxuan Yang*, Beijing University of Posts and Telecommunications, 10 Xitucheng Rd, Haidian Qu, Beijing Shi, 100876, China, yjx@bupt.edu.cn; Haotian Cui*, Tsinghua University, 30 Shuangqing Rd, Haidian Qu, Beijing Shi, China, cht15@mails.tsinghua.edu.cn; Si Li†, Beijing University of Posts and Telecommunications, 10 Xitucheng Rd, Haidian Qu, Beijing Shi, China, lisi@bupt.edu.cn; Sheng Gao, Beijing University of Posts and Telecommunications, 10 Xitucheng Rd, Haidian Qu, Beijing Shi, China, gaosheng@bupt.edu.cn; Jun Guo, Beijing University of Posts and Telecommunications, 10 Xitucheng Rd, Haidian Qu, Beijing Shi, China, guojun@bupt.edu.cn; Zhengdong Lu, DeeplyCurious.ai, Haidian Qu, Beijing Shi, China, luz@deeplycurious.ai.


XXXX-XXXX/2018/7-ART $15.00
https://doi.org/0000001.0000001

## 1 INTRODUCTION

In recent years, web texts including newswires, military texts, electronic medical records and court judgments have been an explosion. Automatically extracting information from plain-text is necessary, which has attracted increasing research in both scientific community and business world. Among various information extraction (IE) tasks, outline extraction is particular and widely applied in many fields like legal online consultation.

Given the textual description of a consultation case, outline extraction requires machine make summary for each individual case by answering several professional questions. Different from formal court statement, question-specific information of informal online cases is often implicitly and cross described among the texts. Thus it is significant to extract pieces of description texts as justification of answers and form multiple question-specific representations.

Traditional IE tasks such as Named Entity Recognition (NER), Relation Extraction (RE) and aspect-based opinion mining have a long history of research. NER aims to identify references to certain types of objects, such as names of individuals, companies and places [34]. RE involves identifying the relationships between entity pairs and representing them as structured triple knowledge. For example, the sentence *"Julia has been working at IBM for three years"* can be described as *"Julia (head) – work for (relationship) – IBM (tail)"*. Compared with our outline extraction task, both these tasks pay attention to lexical and syntactic related discrete information, which relies more on hand-crafted rules but not semantic meaning. Traditional machine learning methods [1, 2, 39] and popular neural network based methods [37, 44] have been presented for these tasks. Aspect is considered as the concept of an object, for which the sentiment opinion is expressed in sentences [38]. It provides remarkable business benefits. For example, retailers identify which characteristics of the product customers are particularly interested in, and get useful feedback by analyzing reviews gathered from websites. Our outline extraction task shares some similarities with this task. The important difference is that the outline extraction focuses on questions which need complicated semantic parsing, but not concepts.

Inspired by recent promising memory-based methods such as Neural Turing Machine [10, 11], Memory Network [41] and EntNet [14], we propose to extract outline of individual cases through a question-specific memory cell (QSMC) network. The network consists of a set of gated RNNs which are sequentially expanded by a set of memory cells. For each cell, there is a key vector $k_j$ representing specific question, and a content vector $h_j$ expressing question related case representation. As texts are sequentially put into the network, question related information will be extracted and used to update corresponding state at each time step by location and content-based independent DNN gating function. A set of task-specific sentence representations are output for downstream applications.

According to [24], rationales as justification of current prediction should be pieces of text that are short and coherent, yet sufficient to represent the semantic information of original texts. So we add a penalization term on loss function to make this regularization.

We develop a novel outline extraction corpus referred as **InjuryCase**, which consists of 3995 real occupational injury cases collected from the online consultation websites. 59 professional questions are provided by three experienced lawyers. Answering these questions can summary the individual case as an outline of 59 answers. We also apply several popular neural network methods to do this task and find that our proposed QSMC network significantly outperforms these baseline methods. As far as we know, this is the first corpus provided for outline extraction.

To summarize, our main contributions are as follows:

- We propose a memory network based outline extraction framework QSMC network to explore a novel outline extraction problem widespread in practice.

- Question-specific information is integrated into gating mechanism to authenticate relevant information and use it to update corresponding sentence states.
- We build a innovative Chinese occupational injury outline extraction corpus using real cases collected from websites. We release this corpus and hope it benefits the research of information extraction.

The rest of the paper is structured as follows. In Section 2, we discuss some popular related work, and in Section 3, we introduce each component of our QSMC network. In Section 4, we provide experimental details and results of occupational injury outline extraction task. In Section 5, we explore another two aspect-extraction tasks to study the robustness of our model. In Section 6, we perform exploratory experiments to demonstrate the effect of gating mechanism and penalization term. Finally, we draw conclusions and put forward some future work in Section 7.

## 2 RELATED WORK

In this section, we discuss the related literature, which is broadly divided into Information Extraction, Memory Network, and Sentence Embedding.

### 2.1 Information Extraction

IE has been a popular text mining techniques recently aims at analyzing plain-texts to identify opinion, information or events which are explicitly or implicitly presented [32]. Main tasks of IE include extracting entity instances and relationships between them such as NER, CO-reference Resolution, RE and Event Extraction [12]. According to the research history, existing methods can be categorized as Knowledge Engineering (KE) and Machine Learning (ML) based methods. Earlier KE methods integrate expertise knowledge and hand-crafted rules into IE systems to recognize related information [28]. The latter ML based methods aim to apply kinds of learning models [13, 30, 35] on these tasks which do not require heavy manual work, which can be divided as supervised, semi-supervised and unsupervised learning fashions. As the renaissance of neural network, deep learning methods have achieved remarkable results on these IE tasks [8, 22, 36, 44]. Inspired by these popular methods, we explore an innovative framework QSMC to resolve a widespread problem in IE named outline extraction.

### 2.2 Memory Network

Recently, memory network seems like a promising direction and have attracted much attention. Memory Network and its variants [3, 11, 41, 42] provide an external memory component to store and update knowledge. A computational module is connected with this part to make it can be read and written. The difference between our model and their work is that we integrate the sophisticated controller network into the memory cells to extract, store and reason on-the-fly as it reads texts. Location-content based addressing mechanism is also a significant improvement adapting to our multi-aspects extraction task. The Dynamic Memory Network proposed in [43] is similar to our work, but it sequentially updates the hidden states via Softmax gating function, while our method does it in parallel.

Our work is also closely related to EntNet [14] in the read-write operations. However, the EntNet framework is a story understanding scenario to explore word state representations and make prediction according to the question. Parameters are shared among RNNs to explore invariant laws of the world. However, in our framework, memory cell is modified to adapt to our multi-aspects extraction problem. Parameters are private for each kind of cell to make the aspect-related representations do not interfere each other.

### 2.3 Sentence Embedding

As much progress has been made in word embedding learning, sentence representation still needs to be explored. Existing methods can be categorized as either unsupervised or supervised style. Methods like SkipThoughtvectors [20], ParagraphVector [23] and FastSent [15] are explored to train universal sentence embeddings by large sentence corpus in an unsupervised way like the method of word2vec [31]. Although taking advantage of huge unlabeled corpora, general sentence embeddings usually perform worse than specific ones trained with supervision in certain task.

Supervised methods have been broadly researched recently by a variety of recurrent networks [16], convolutional networks [17, 18] and recursive networks [40]. They directly use last state, take max (or average) pooling operation over a set of hidden states or combine other linguistic features [27, 33] to get a distributed vector representation for sentences. These methods perform well to train a specific representation for each task, but not suitable to obtain multi-aspects representations simultaneously. It is also relatively hard to capture semantic meanings by simple max or average operation.

Some attention mechanism based methods are also proposed recently to calculate sentence representations as a weighted sum of hidden states at each time step [4, 25, 26]. Our model is different from these works as we build an independent processor for each task, so aspect-relevant information is stored and updated isolate. However, in attention-based methods, output hidden states at each time step are fixed, and different representations are computed only by changing weights. Penalization term is combined in our loss function to restrict the task related descriptions are short and continuous.

## 3 MODEL

In this section, we describe QSMC framework in detail. Firstly, we give the task definition. Secondly, we introduce three main parts: **(a) Encoder (b) Question-specific Memory Cell** and **(c) Output Layer** as shown in Fig. 1. Thirdly, we describe how to transform the input into proper form according to downstream applications like classification and regression. At last, we formulate the penalization term which is added at the end of loss function.

### 3.1 Task Definition

Given a length-variable case text $X = \{x_1, x_2, \ldots, x_n\}$ consisting of n words and a set of professional questions, we aim to sequentially process text, identify question related information and update question-specific sentence representations on-the-fly. A set of numeric answers will be output as $Y = \{y_1, y_2, \ldots, y_m\}$, according to the downstream applications. In our framework, we use pre-trained distributed word embedding matrix $L \in \mathbb{R}^{d \times |V|}$ to initialize input texts, where $d$ is the dimension of word vector, and $|V|$ is vocabulary size.

### 3.2 Encoder

The encoder layer transforms the input sequence of words into distributed vector representations to express sentence semantic meaning. According to the sequential characteristic of our framework, standard RNN component and its many variants can be free chosen as the encoder.

In our work, we mainly explore two fashions of RNN as gated recurrent units (GRU) [5] and bi-directional GRU (BiGRU).

*3.2.1 For GRU Fashion.* GRU is proposed as a sequence modeling mechanism by recurrent applying a transition function to its internal hidden state vector $h$ and address the problem of exploding or vanishing gradients [21] in a simpler way than long short-term memory (LSTM) network [16].

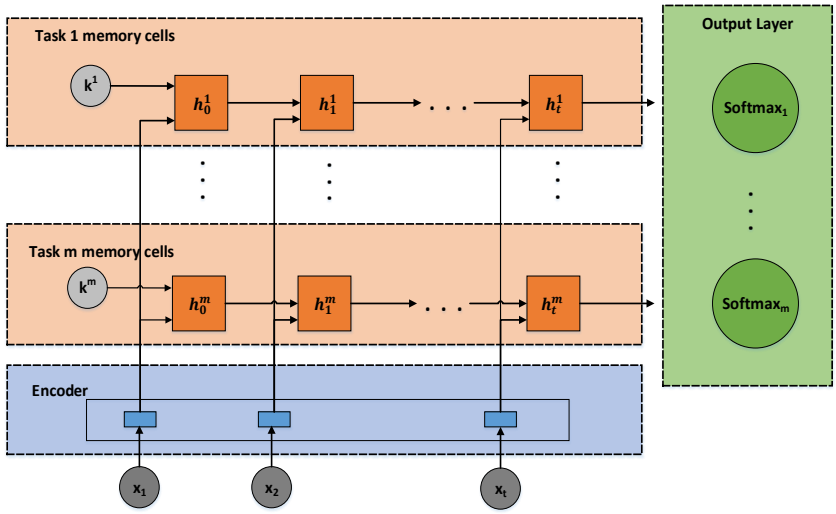



Fig. 1. Framework of Question Specific Memory Cell network. It includes three parts as (a) Encoder layer (b) Question-specific memory cell layer (c) Output layer.

In order to encode input text words at each time step and update current sentence state, update and reset gates are defined as follows:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \tag{1}$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \tag{2}$$

$$\widetilde{h_t} = \phi(Wx_t + [U(r_t \odot h_{t-1})]) \tag{3}$$

$$h_t = (1 - z_t)\widetilde{h_t} + z_t h_{t-1} \tag{4}$$

where $x_t$ is input at current step, $\phi$ denotes the logistic sigmoid function and $\odot$ means element-wise multiplication. The reset gate $r$ decides whether the previous state is no longer relevant and should be drop out. The update gate $z$ controls how much the past state matter now and helps eliminate vanishing gradient problem.

*3.2.2 For BiGRU Fashion.* Existing research [7] has demonstrated that for many natural language processing tasks like sequence labeling, it is beneficial to obtain both past and future contexts. In our outline extraction task, the answers of different questions are cross-introduced and supported by each other. For example, a certain disability level description will help the consultant know that the injury has been authenticated by the relevant institution. Thus, it is significant to combine forward and backward features together to identify whether current input is related to the specific question.

We utilize a BiGRU network to present the input sequence forwards and backwards. Then the two hidden states are concatenated as final output, like the bi-directional LSTM proposed in [9].

### 3.3 Question Specific Memory Cell

QSMC layer aims to model multiple sentence representations attend to different aspects. It can be seen as a bank of gated recurrent networks, which are parallel expanded by independent GRU cells along each time step. Each cell has a key vector $k^j$ represents the semantic meaning of the question

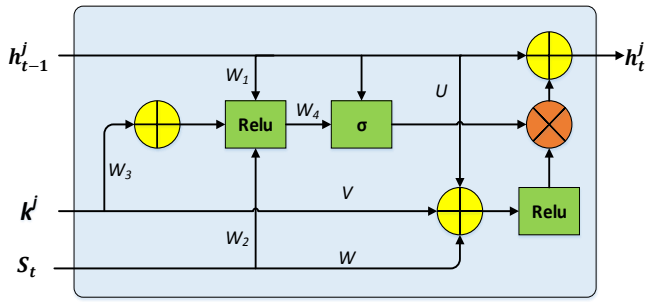



Fig. 2. QSMC architecture. $k^j$ and $h^j_{t-1}$ are two properties for each cell and $s_t$ is the input at current step. Gating mechanism is represented as a neural network form. Question related sentence states are updated by these three terms.

and a hidden state vector $h^j$ represents the question-specific representation for this sentence. The state vector $h^j$ updates according to a location and content-based gating mechanism for which the structure is shown in Fig. 2.

*3.3.1 Read Operation.* In the gating function, key vector $k^j$, question specific representation $h^j$ and input vector $s_t$ are organized as a two-layer neural network to explore semantic similarity among them. The component $W_3 k_j$ can be seen as a location-term to figure out whether current input is a description about the certain question of this memory cell. While $W_1 h^j_{t-1}$ as a content-term is used to measure whether input context is semantically related to the sentence state about this question.

The content-term is designed based on human narrative habit that the description should be coherent. Once the target word is identified by location-term, the nearby context information will be manipulated by content-term to make the sentence representations more complete.

The description identify process can be seen as a memory read operation. Our gating function is given as:

$$g^j_t = \sigma[\phi(W_1 h^j_{t-1} + W_2 s^T_t + W_3 k_j + b_1) + b_4] \tag{5}$$

*3.3.2 Write Operation.* After the reading operation has authenticated related question of the current input, the corresponding memory state should be updated synchronously. The gating score decides to what extent the new candidate should be used to update cell state, as a write operation. Related update equations are:

$$\widetilde{h^j_t} = \phi(U^j h^j_{t-1} + V^j k^j + W^j s_t) \tag{6}$$

$$h^j_t = h^j_{t-1} + g^j_t \odot \widetilde{h^j_t} \tag{7}$$

$$h^j_t = \frac{h^j_t}{\|h^j_t\|} \tag{8}$$

where the function $\phi$ can be chosen as any activation function. In our experiment, we use sigmoid function. $U, V, W$ are three trainable parameters used to compute candidate inner representations, and they are independent among different cells to produce candidates respectively. The final normalization step can be seen as rewrite process to forget some old information and emphasize some

new information in their phase. This operation also prevents any potential explosive growth caused by successive catastrophic gating scores. The output of expert cell layer is the concatenation of last time step hidden states $h_1, h_2, ..., h_m$, where $m$ presents the number of questions.

### 3.4 Output Layer

Question-specific sentence representations output from QSMC layer are fed into the output layer. Our goal here is to make the vectors are sufficient to express question related semantics of this sentence, which means the they should produce gold results according to downstream applications.

For classification problem, the output layer composed of a set of independent $Softmax$ functions, and we train the model by minimizing the classification cross entropy error as follows:

$$L = \sum_{h \in T} \sum_{j} P_j(h) \cdot \log(y_j) \tag{9}$$

$P_j(h)$ is the predicting probability of vector $h$ with task $j$, and $y_j$ is 1 or 0 indicating the correct label of task $j$ for this sentence. $T$ means all training instances, and $j$ means the $j-th$ task.

For regression problem, the corresponding mean square error (MSE) loss function is given as:

$$L = \sum_{h \in T} \sum_{j} ||f_j(h) - y_j||_2^2 \tag{10}$$

$f(h)$ is the predicting score for task $j$ and $y_j$ is the ground-truth score.

In our outline extraction problem, the answer of each question is a discrete label as [0,1,2]. We take cross entropy form in this task. While for some sentiment analysis problem, the sentimental polarity is given in fractional score. In this way, MSE form maybe more suitable.

### 3.5 Penalization Term

The gating mechanism in QSMC corresponds to semantic correlations between input text and questions. Inspired by the work in [24], we want to guide the gating mechanism to select short and coherent texts. It is realized by choosing few words, and the words are consecutive rather than isolated. We introduce a penalization term for this goal as:

$$P(g) = \lambda_1 \|g\| + \lambda_2 \sum_{t} |g_t - g_{t-1}| \tag{11}$$

where the first term computes L1 form of the whole gating values in the text to reduce the number of words and the second one encourage continuity of selected words by making gating score of continuous words change slow. The penalization term is added to original objective function. Our objective cost function is expressed as follows:

$$cost = \sum_{j} [L + \lambda_1 \|g\| + \lambda_2 \sum_{t} |g_t - g_{t-1}|] \tag{12}$$

Note that the penalization term we put forward here is a general format. It should be adapted to different tasks according to practical requirement by adjusting coefficients $\lambda_1$, $\lambda_2$. For example, if the sentence is not long enough, the keywords may be few and not consecutive. So the second term is not reasonable and should be ignored.

## 4 EXPERIMENTS ON OCCUPATIONAL INJURY OUTLINE EXTRACTION

We mainly study the performance of the QSMC model on our novel occupational injury outline extraction task.

Table 1. *Name, Description* and *Answer Category* statistics of nine pre-defined occupational injury ontology problems.

| PROBLEM | Description | Answer Category |
|---|---|---|
| InjuryIden | Whether or not the occupational injury has been identified? | 3 (yes/no/not mention) |
| Employ | Do you have employment relationship with company or individual? | 3 (yes/no/not mention) |
| ConfirmLevel | Does the occupational injury have been graded by authority? | 3 (yes/no/not mention) |
| ApplyPay | Do you ask the process of how to apply for occupational injury compensation? | 2 (yes/no) |
| WorkTime | Does the injury happen during work hours? | 3 (yes/no/not mention) |
| WorkPlace | Does the injury happen in workplace? | 3 (yes/no/not mention) |
| AssoPay | Do you ask for the amount of compensation? | 2 (yes/no) |
| HaveMedicalFee | Is there any medical fee in your case? | 3 (yes/no/not mention) |
| Identity | What is your relation with the injured? | 4 (the injured/the injured families/employer/others) |

### 4.1 Dataset Construction

*4.1.1 Data Acquisition and Question Institution.* To construct a proper online construction corpus, we collect the raw data of occupational injury cases from the website FindLaw[1], which serve for the public to seek legal advice from professional lawyers. After crawling, we de-duplicate the raw data and filter out some noisy and incomplete cases. Finally, 3995 regular and representative samples are adopted.

According to the case statistics, we decide to focus on two problems which are consulted most frequently as "Is this case can be authenticated as occupational injury?" and "How much compensation can I get for such an injury?" In order to provide professional advice for these two problems, three professional lawyers organize 59 relevant questions which list some establishment conditions of these two problems. Answering these questions will provide sufficient information to make a professional judgment and generate advice. Since there are many ones distributed unbalanced which will cause some deviation, we pick nine relatively balanced distributed questions which are introduced in Table 1.

For example, in Fig. 3, the case *"I work at a construction site and suffer from occupational injury. Authority identifies the injury level is ten. I want to know how much I can be compensated for."* is provided by an injured. Outline extraction maps this text description into 9 labels as shown in the right box.

[1] http://china.findlaw.cn/.

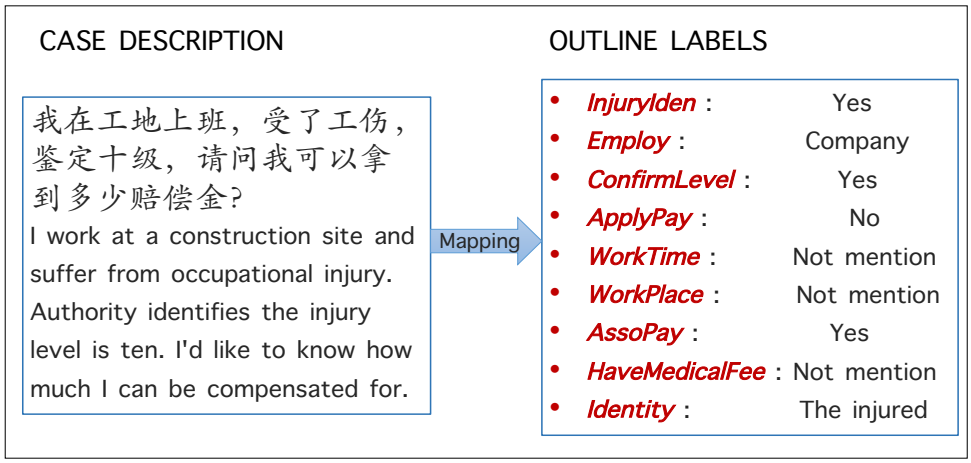



Fig. 3. Example of mapping Occupational Injury Case to ontology labels. The case description in left box is mapped as nine labels (*Yes, No, Not mention*) in right box to answer related problems.

*4.1.2 Annotation Procedure and Statistics.* For most of these questions, the answer can be categorized as yes/no/not mention as shown in Table 1. But the last question aims to figure out the relation between the case descriptor and the injured, which compose of four types as **the injured, the injured families, employer** and **others**.

In order to guarantee the annotation quality, we recruit three experts who are professional at occupational injury to annotate 300 cases as templates. When the disagreement appears, they discuss and make final decision according to the majority rule. Then we make this task as a crowdsourcing annotation task. Ten annotators are taught detailed rules to annotate the other 3695 cases. After repeated check and revision for three times, the corpus achieves the inter annotator agreement of 95.3%, which is enough highly qualified to be released as a corpus.

The final annotation statistics of these questions are given in Fig. 4. We can see the annotations are distributed unbalanced. The label **Not Mention** accounts for absolute proportion because most people do not introduce the specific information when they describe the situation. This part need to be further figure out by subsequent QA.

### 4.2 Comparative Methods and Training Details

The proposed approach is compared with three recent baselines include:

- **Self-attention** [25]: is applied on the LSTM layer to measure different weights for hidden states focusing on current aspect. The sentence representations then will be presented as a 2-D matrix for which each row is attending to a different part of the sentence.
- **CNN** [18]: is proposed by Kim, which innovatively use convolution operation to extract semantic features and use an independent Softmax function for each independent classification task.
- **BiGRU max-pooling** [6]: extends original GRU approach as a bi-directional network composed of a forward GRU and a backward one. The max-pooling operation is also done among each time step output vectors to get the final state.

In our experiment, word vocabulary is generated from this corpus directly which contain 7.3K words. 100-dimensional word embeddings are initialized randomly and tuned during the training

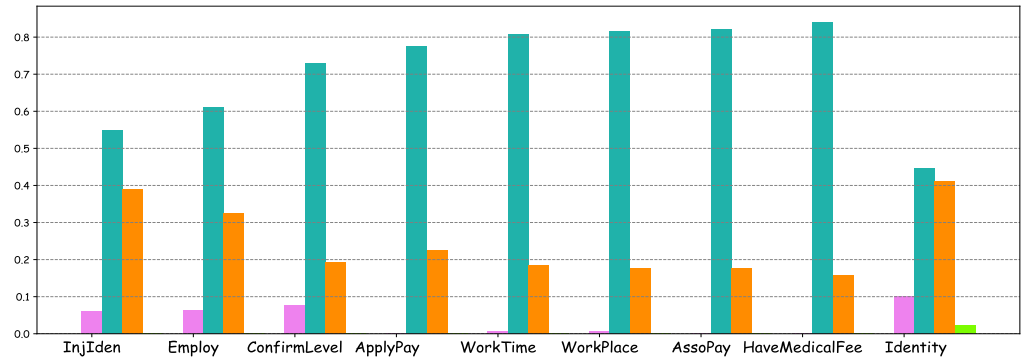



Fig. 4. Annotation statistics of the selected questions. For the first eight questions, the pink bar means **No**, the blue bar means **Not Mention** and the orange bar means **Yes**. For the last question *Identity*, the pink bar represents **The injured Families**, the blue bar represents **The Injured**, the orange bar represents **Employer** and the green bar represents **Others**.

Table 2. Evaluation results on Occupational Injury Outline Extraction Task performed by three baselines and QSMC approaches.

| **MODEL** | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1-score*** | ***Parameter*** |
|---|---|---|---|---|---|
| Self-attention [25] | 0.762152 | 0.661111 | 0.76222 | 0.702222 | 2608449 |
| BiGRU-max pooling [6] | 0.760978 | 0.66 | 0.761111 | 0.698889 | 3575229 |
| CNN [18] | 0.787514 | 0.665556 | 0.785556 | 0.718889 | 5180949 |
| GRU QSMC | **0.785223** | **0.72** | **0.785556** | **0.737778** | 3771050 |
| BiGRU QSMC | **0.791714** | **0.74333** | **0.79** | **0.755556** | 3711050 |

process. GRU model uses 100 dimension hidden states. BiGRU-based model utilizes a two independently GRU with 50 dimensions in opposite directions respectively. BiGRU-max pooling model then performs max pooling operation across hidden vectors at each time step to get the sentence vector. For all the baselines and our approach, we use a 2-layer *tanh* output MLP with 1000 hidden states to get the classification results. 50% dropout is set on the first MLP layer during training and use Adam optimizer with a learning rate of 0.0005, batch size 64. We also clip the norm gradients to be between -0.5 and 0.5.

### 4.3 Main Results

We use **accuracy, precision, recall**, and **F1-score** as measurements. Average majority accuracy of these nine problems is 0.7107444. Results of our approach against other methods are listed in Table 2. We can see that our approach performs remarkably well with proper parameter quantity. Our approach uses BiGRU as encoder performs better for which F1-score is 0.755556 than one directional GRU encoder because the memory cell can look at both forward and backward information at the same time to calculate content similarities. The hidden state dimension of BiGRU is taken as half of GRU.

Table 3. Mean Square Error results of all baselines and QSMC approaches for BeerAdvocate Task.

| MODEL | *Mean Square Error* | *Epoch Number* |
|---|---|---|
| BiGRU | 0.0205 | 20 |
| RCNN [24] | 0.0225 | 21 |
| GRU | 0.0198 | 20 |
| Feature SVM | 0.01908 | 80 |
| CNN | 0.0165 | 26 |
| Self-attention [25] | 0.0161957 | 5 |
| GRU QSMC linear | 0.0159237 | 10 |
| BiGRU QSMC DNN | **0.0154035** | 4 |
| GRU QSMC DNN | **0.0149953** | 5 |

## 5 EXPERIMENTS ON ASPECT EXTRACTION TASK

Existing aspect extraction models mainly pay attention to tasks with only one target aspect. However, in practice, we often need to extract multi-aspect related information synchronously. Fortunately, our QSMC framework can be easily extended to solve this problem. Each cell can be used to manipulate the information about a specific aspect. In this section, we apply our model on other two multi-aspect extraction tasks as: (1) Beer Advocate: multi-aspects sentiment analysis; (2) Wine Exploration: variety recognition and scoring.

### 5.1 Beer Review Multi-aspects Sentiment Analysis

We use Beer Reviews dataset[1] which has been used in prior work [24, 29]. It contains 1.5 million reviews written by customers describing five aspects of the beer including **appearance, aroma, palate, taste** and **overall**. Five fractional scores are also given in range [0, 5] to evaluate these aspects for each review. In our experiment, we choose the subset present in [24], which is extracted from the original complete dataset to ensure the aspects are less correlated. Scores are normalized to [0, 1] to make a accurate supervision for regression. The training data is composed of 70k reviews while the development set contains 10k reviews.

Besides the baselines introduced in Section 4.2, we also perform another two comparative methods for this task as:

- **Feature-based SVM** [24]: traditionally performs state-of-the-art on aspect level sentiment classification. We compare with a common system using n-gram features and TF-IDF features.
- **RCNN** [24]: proposed by Tao to extract pieces of text as rationales to explain predict scores of five aspects.

Since it is a regression problem, we take Mean Square Error (MSE) as measurement. We use 200-dimensional word embeddings pre-trained on review and Wikipedia as initialization, and tune them during the training process. We train our approach with back-propagation. The gradient-based optimization is performed by Adam update rule [19]. We take batch size as 128. All parameters are initialized by the Gaussian distribution and updated by a learning rate of 1e-4 which decayed by 0.95.

[1] www.beeradvocate.com.

Table 4. Statistics of wine variety classification accuracy and scoring MSE results performed on three baselines and QSMC approach.

| **MODEL** | ***Variety Classification Accuracy*** | ***Scoring MSE*** |
|---|---|---|
| GRU | 0.429487 | 0.0144743 |
| BiGRU-max pooling [6] | 0.430889 | 0.0166195 |
| BiGRU | 0.452324 | 0.0168615 |
| BiGRU QSMC | **0.485978** | **0.0121839** |

Results are summarized in Table 3. We can see that our QSMC approach with GRU encoder performs rather well and need to be trained by fewer epochs. We also investigate the influence of gating mechanisms in linear or nonlinear forms. Results show that 2-layer DNN gating function performs better than linear form, for which the MSE is 0.0149953. So we can conclude that a sophisticated gating function can calculate content similarity with aspects better. Our approach outperforms all of the baselines, which demonstrate the effectiveness of our method.

### 5.2 Wine Variety Recognition and Scoring Task

At last, we use the wine review dataset given in Kaggle Competition to demonstrate our approach. The set contains 150k wine reviews include ten fields about the wine as **Points, Title, Variety, Country, Province, Region1, Region2, Winery, Designation, Price, Taster Name** and **Taster Twitter Handle**. After downloading, we remove duplicate and drop all data lack price or description. Sentences have less than 20 words are also removed. Choosing data whose **variety** attribute is among ten distributed balanced categories, we finally get 29920 regular wine reviews. We take 20000 instances as the training set, 5000 instances as the development set and other 4920 ones as the testing set.

According to the assignment in competition, we predict wine variety and score simultaneously according to the descriptions given by sommeliers. The variety prediction is realized as a classification problem while the score prediction is a regression problem. Original points are given in range [80, 100], and we normalize them to [0, 1] and use them as the only supervision for regression. The final loss function is a weighted sum of classification cross entropy error and regression mean square error.

We also compare our approach with three baselines mentioned in Section 4.1. Word embeddings are initialized as the same with the BeerAdvocate task. A 2-layer MLP with 2000 hidden states are used in classification problem and for regression problem, 30 hidden states. Results summarized in Table 4 show that our approach makes a remarkable improvement compared with all baselines.

### 5.3 Model Versatility Analysis

Given that textual information extraction from different perspectives at the same time is very common in practice, previous two aspect-extraction tasks have effectively demonstrated the robustness, versatility and generalization capacity of our framework. The key lies in that we allocate a distinguished sign vector for each sub-task to let the information find its own location precisely. The vector has different meanings when the framework is applied to various information extraction tasks. For our outline extraction problem, the vector presents semantic meaning of the specific question. While for aspect extraction task, it can be seen as the concept meaning of corresponding

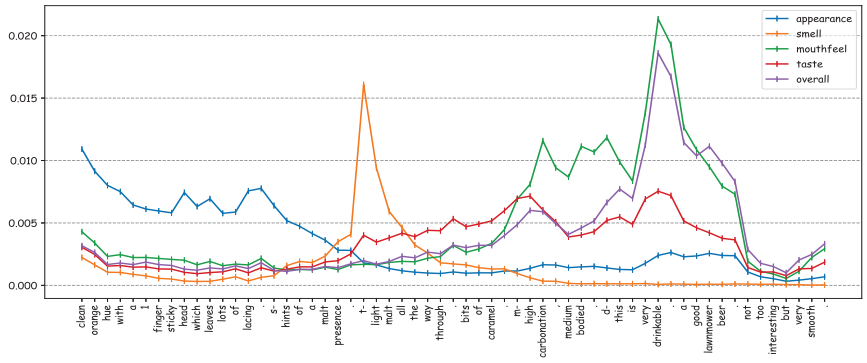



Fig. 5. Aspect related gating values for each word. Y-axis represents the gating score, while x-axis represents the input words in chronological order. Higher score means the gate is open for this word with this aspect.

aspect. We can also impose strong structural priors from this vector by some well-designed initializations, which is a significant attempt to guide the network reading texts based on common sense.

## 6 EXPLORATORY EXPERIMENTS

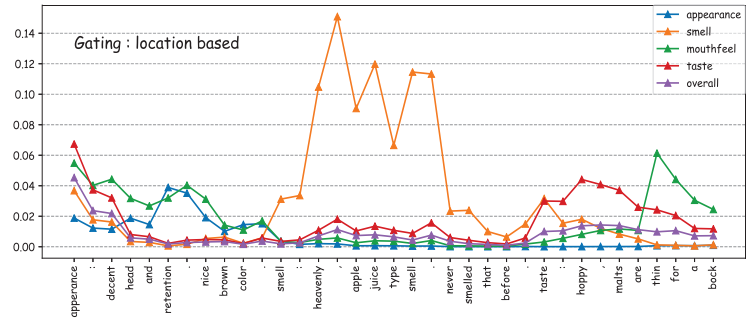
In this section, we conduct two qualitative exploratory experiments to study the effects of gating mechanism and penalization term.

### 6.1 Effect of Location and Content-based Gating Mechanism

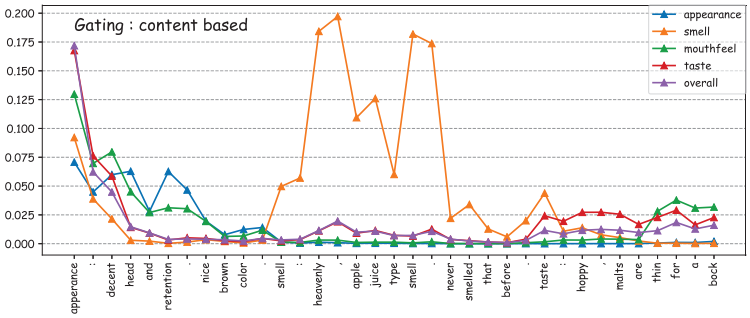
The gating mechanism is particularly designed to select useful texts word by word for different questions. To directly understand what happened, we plot the gating values of the input word at each time step .

We randomly select a review from the validation set of the BeerAdvocate task as *"clean orange hue with a 1 finger sticky head which leaves lots of lacing. s- hints of a malt presence. t- light malt all the way through. bits of caramel. m- high carbonation, medium bodied. d- this is very drinkable. a good lawnmower beer. not too interesting but very smooth."* The lines plotted with different colors in Fig. 5 represent gating values of each word correspond to different aspects. Higher gating value means the gating mechanism think current input is closely related to this task. So we can find the gating lines of **"appearance"** and **"smell"** are higher at words corresponding to these aspects precisely, which demonstrate the effectiveness of our gating mechanism. But for the other three aspects, the curves are correlative to a certain extent since the words used to describe **"taste"** and **"mouthfeel"** are similar and the words used to describe **"overall"** is not apparent.

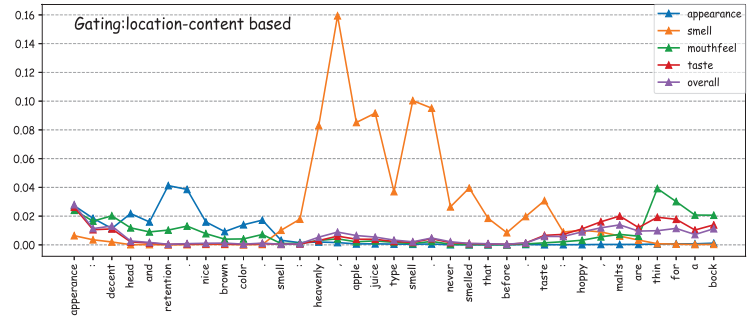
We also explore the effect of location and content addressing method in gating function. In Fig. 6, gating lines of the same sentence are plotted when gating function is composed of different terms. We can see that location and content-based gating mechanism choose more precise words and make gating lines for different aspects separated further. Particularly, **"appearance"** related texts cannot be selected clearly by location-based or content-based gating functions as shown in first two pictures. But in the third one, these words are chosen apparently. We perform BiGRU QSMC framework on Occupational Injury and Wine Recognition problems when gating functions vary as three forms. Results are illustrated in Fig. 7. From these two figures, we can tell that the gating component performs best with location and content-based form. Since these two datasets are relatively small compared with the BeerAdvocate corpus, gating lines cannot be plotted clearly.




(a) Location-based




(b) Content-based




(c) Location and Content-based

Fig. 6. Gating visualization of the same sentence with 5 aspects, and different pictures means gating mechanism use different forms. (a) (b) The gating mechanism is only with location addressing method or content addressing method. (c) With both location and content-based method.

## 6.2 Effect of Penalization Term

Adding penalization term on original objective function is expected to make selected texts to be more proper. It makes sense to evaluate how significant the improvement can be brought by this

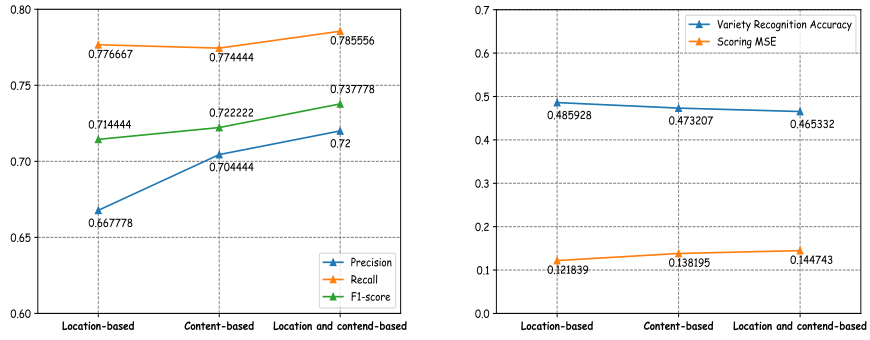



Fig. 7. Average precision, recall and F1-score of Occupational Injury problem (left) and MSE of Wine Exploratory task (right) with different gating mechanisms.

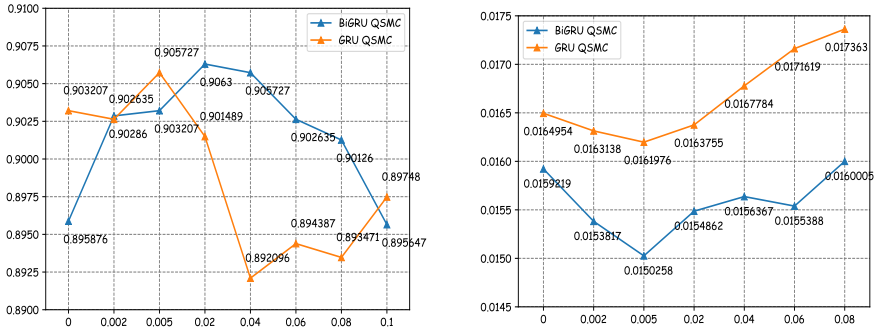



Fig. 8. Average accuracy of Occupational Injury problem (left) and MSE of Beer Review (right) when various coherent efficient are taken in penalization term with GRU QSMC and BiGRU QSMC framework.

component and explore a proper coefficient. We explore two models for both BeerAdvocate Reviews task and Occupational Injury Case task as GRU QSMC and BiGRU QSMC.

We vary sparsity coefficient $\lambda_1$ from 0 to 0.1 for each task and set the coherent coefficient $\lambda_2$ as $\lambda_2 = 2 \cdot \lambda_1$. Penalization term is removed when lambda1 is taken as 0. Average classification accuracy and average MSE for these two tasks are plotted in Fig. 8.

From this Figure, we can find that these models perform best when sparsity coefficient is 0.005 or 0.02. The model performs quite differently with respect to variable coefficient. Both the classification accuracy and MSE measurements vary more than 1%.

## 7 CONCLUSIONS

In this paper, we present an question-specific memory cell network for outline extraction task. A location and content-based gating mechanism is provided to extract question-related information and learn sentence representations. Results show that this approach makes a significant improvement over other traditional machine learning methods and some state-of-the-art neural network

approaches. There are also some limitations in this approach. In many cases, the information we want to extract from texts is interrelated. We can infer the information of one aspect from other aspects even if it is not described in text. So the zero-short problem is effectively relieved.

In future work, we would like to investigate this problem by adding some prior knowledge in key vector and exploring some attention mechanisms among memory cells to make them learn task relations automatically.

## REFERENCES


[1] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 8–15, 2003.

[2] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Description of the mene named entity system as used in muc-7. 1998.

[3] Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesauro, and Yoshua Bengio. Hierarchical memory networks. *arXiv preprint arXiv:1605.07427*, 2016.

[4] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. 2016.

[5] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Computer Science*, 2014.

[6] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[7] Chris Dyer, Miguel Ballesteros, Ling Wang, Austin Matthews, and Noah A. Smith. Transition-based dependency parsing with stack long short-term memory. *Computer Science*, 37(2):321–332, 2015.

[8] Yoav Goldberg. A primer on neural network models for natural language processing. *Computer Science*, 2015.

[9] Alex Graves, Abdelrahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks," icassp. 38(2003):6645–6649, 2013.

[10] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *Computer Science*, 2014.

[11] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[12] Ralph Grishman. Information extraction. *IEEE Intelligent Systems*, 30(5):8–15, 2015.

[13] Leong Chieu Hai and Hwee Tou Ng. Named entity recognition: a maximum entropy approach using global information. In *International Conference on Computational Linguistics*, pages 1–7, 2002.

[14] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. *CoRR*, abs/1612.03969, 2016.

[15] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. pages 1367–1377, 2016.

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

[17] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *Eprint Arxiv*, 1, 2014.

[18] Yoon Kim. Convolutional neural networks for sentence classification. *Eprint Arxiv*, 2014.

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *Computer Science*, 2015.

[21] J Kolen and S Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*. Wiley-IEEE Press, 2007.

[22] Shantanu Kumar. A survey of deep learning methods for relation extraction. 2017.

[23] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. 4:II–1188, 2014.

[24] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. 2016.

[25] Zhouhan Lin, Minwei Feng, Cicero Nogueira Dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. 2017.

[26] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional lstm model and inner-attention. 2016.

[27] Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. Dependency-based convolutional neural networks for sentence embedding. pages 174–179, 2015.

[28] Monia Mannai, Wahiba Ben Abdessalem Karâa, and Henda Hajjami Ben Ghezala. Information extraction approaches: A survey. 2018.
[29] Julian Mcauley, Jure Leskovec, and Jurafsky Dan. Learning attitudes and attributes from multi-aspect reviews. pages 1020–1025, 2012.
[30] Andrew Mccallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. of the Conference on Computational Natural Language Learning*, pages 188–191, 2003.
[31] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Computer Science*, 2013.
[32] Raymond J. Mooney and Razvan Bunescu. Mining knowledge from text using information extraction. *Acm Sigkdd Explorations Newsletter*, 7(1):3–10, 2005.
[33] Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. Discriminative neural sentence modeling by tree-based convolution. 2015.
[34] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):págs. 3–26, 2007.
[35] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):págs. 3–26, 2007.
[36] Thien Huu Nguyen and Ralph Grishman. Combining neural networks and log-linear models to improve relation extraction. *Computer Science*, 2015.
[37] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *The Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, 2015.
[38] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. *Aspect extraction for opinion mining with a deep convolutional neural network*. Elsevier Science Publishers B. V., 2016.
[39] Satoshi Sekine. Nyu : Description of the japanese ne system used for met-2. 1998.
[40] 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.
[41] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *Computer Science*, 2015.
[42] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.
[43] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International Conference on Machine Learning*, pages 2397–2406, 2016.
[44] Lin Yao, Hong Liu, Yi Liu, Xinxin Li, and Muhammad Waqas Anwar. Biomedical named entity recognition based on deep neutral network. *International Journal of Hybrid Information Technology*, 8, 2015.