國 立 清 華 大 學

碩 士 論 文

利用語意嵌入常識模板與蒙地卡羅樹搜尋法產生連貫性文章

Generate coherent text using semantic embedding, common sense templates and Monte-Carlo tree search methods

系所別：資訊工程學系碩士班

學號姓名：104062584 陳櫻仁 (Ying-Ren Chen)

指導教授：蘇豐文 博士 (Prof. Von-Wun Soo)

中 華 民 國 110 年 1 月

摘要

自動產生相關且可懂的文字是一項困難的研究，很多現存的自然語言生成系統沒有考慮到詞和詞之間的關係，且沒有理解常識知識。詞和詞或句子間的關係是超出預期的緊密結合，因此缺乏理解常識知識的語言生成系統往往會造出非預期中的段落或句子。為了改進這個問題，我們從 ConceptNet 裡自動抓取常識知識並結合詞嵌入及深度神經網路篇章連貫性模型到蒙地卡羅搜尋樹裡，在極大的搜尋空間及有限的時間資源裡來尋找次佳解，最後根據使用者給定的初始詞來生成具連貫性的段落。我們也觀察到以統計記數方法而建成的詞嵌入經過調整後，在相似及相關性的任務上比用預測來建成的詞嵌入模型還要精準。我們的詞嵌入在相關性任務上得到了 0.679 的 Spearman 分數，勝過了其他預訓練的詞嵌入。我們最後採用人工來評測文字連貫性，評測結果為採用篇章連貫性模型及修改擴增後的 ConceptNet 所生成的文字更具有連貫性。

關鍵詞: 自然語言生成、ConceptNet、常識知識、詞嵌入、蒙地卡羅樹搜尋法

Abstract

The objective of our research was to generate a coherent, understandable text which is a challenging task. Many of current natural language generation systems that based on word appearance frequency didn't consider the relations between words and lack understanding of commonsense knowledge. The relations between words or sentences combined so closely and subtlety that are often beyond the system's expectation. Therefore, unexpected paragraphs or sentences may be generated that leads to the incoherence of the generated text. To remedy this problem, we extracted commonsense knowledge templated from ConceptNet automatically. We combined a constructed word semantic embedding model and a designed Deep Neural Network of discourse coherence model with Monte-Carlo Tree Search to find suboptimal branches in a large search space and the limited time. Our system can generate a more coherent paragraph given user's input concept. We also observed that with proper techniques, count-based word embedding can perform better than prediction-based one on similarity/relatedness tasks. We get 0.67 Spearman's score on relatedness task which outperforms other pre-trained word embeddings. We eval-

uated generated paragraphs by human rating, our model can generate more coherent paragraphs when using the discourse coherence model and refined ConceptNet.

Keywords: Natural Language Generation, ConceptNet, commonsense knowledge, word embedding, Monte-Carlo Tree Search.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural language generation (NLG) can be viewed as an automated process that can convert a text or data into some targeted text, namely text-to-text or data-to-text. Text-to-text generation usually takes some existing text as input and outputs a coherent, understandable, well-formed and fluent text automatically. It can be applied in machine translation, document summarization or text paraphrasing. Data in data-to-text generation can be structured knowledge base (KB), numerical data or an image. It can be applied in robo-journalism, weather forecast or image captioning.

Some researches used structured commonsense knowledge base (CKB) to generate natural language [1–4]. However, it exists some potential problems in CKB. In [4], they used ConceptFlow, one-hop and two-hop concepts of node, to traverse knowledge graph to generate dialogues. If the data in knowledge graph is inaccurate or erroneous, the traversed concepts would be wrong especially for the two-hop concepts. The distance between N-hop concepts and original concept is inversely proportional to relatedness. The errors will be amplified as the distance increasing. Consequently, an accurate and well-formed CKB plays an important

role in NLG.

However, constructing an accurate and well-formed CKB is usually time-consuming and expensive. There is a trade-off between coverage and correctness. ConceptNet [5] is a large CKB which contains a large number of errors, such as typos, relations in reverse order or meaningless data, etc. We refine Chinese ConceptNet to make sure the commonsense knowledge in it is correct so that generated text has higher quality. Furthermore, the main language in most NLG tasks is English. There are relatively few researches that focus on the Mandarin Chinese NLG. Because the Mandarin Chinese isn't a major language in the world and Chinese ConceptNet also has a smaller size than other languages (in fact, Chinese ConceptNet ranks the 10th among other core languages in terms of size.) Therefore, we expand Chinese ConceptNet by plesionyms (near-synonyms) to prevent the generated text from dominating by some frequent concepts.

One of the vital tasks in NLG is how to select appropriate words so that generated text is coherent. However, since the word space is very huge, the Monte-Carlo Tree Search (MCTS) [6] method is a heuristic search algorithm for finding optimal decisions by random sampling in finite time or resources. It has been recognized for its successful in the Go game [6] compared to traditional search algorithms. The average branching factor is 250 in GO game, and the average moves in one game is approximately 150. The search space can be 10^{360} or even larger depending on the moves. It's impossible to exhaustive search in such a large space. Use MCTS to random sampling the possible actions until reaching leaf nodes, evaluate the result and back propagate to the current node. It can estimate the expected gain of specific position in board. AlphaGo [7] used deep

reinforcement learning and MCTS to beat professional world champion Go player in 2016.

It has been proved that machine can take optimal or suboptimal moves in a finite time (GO game has fixed thinking time) and large search space using MCTS. Both Go game and NLG can't evaluate results until the last step, and they both have a large search space (Go game is still larger than NLG tasks). A paragraph consists of words, and each word has large number of possible choices. We can't conduct exhaustive search and evaluate all possible paths. Some researches has been applied MCTS in NLG tasks [8–11].

One major difference between GO game and NLG tasks is that GO game has a clear binary result, win or lose. However, it's vague and ill-defined to evaluate whether a generated text is coherent or not. It's subjective to human, different persons may have different judgements depending on their own perspectives and intuitions. To address this problem, we attempt to evaluate the quality of a paragraph automatically and objectively by training a deep neural network in large text corpora.

To sum up, we propose a NLG model which can select concept words by the MCTS, combining text by human-crafted templates and use a deep neural network (DNN) architecture to evaluate and ensure the coherence of the generated Chinese texts.

Thesis Organization

Chapter 2 is the overview of some background knowledge regarding NLG tasks, KB, concept association and discourse introduction. Section 2.4 list and discuss some related work about NLG tasks. In chapter 3, we describe the proposed methods to refine and expand Chinese ConceptNet, and use it to generate and evaluate coherent paragraph. We then evaluate the models or methods we proposed with some experiments in chapter 4, and conclude our research in chapter 5. The extra data and information are provided in appendix.

Chapter 2

Background

NLG relies on CKB, word embedding and generated system. A paragraph consists of sentences, and a sentence consists of concepts or words. Plain text is a bunch of symbols if it can't be understood by machine. Word embedding is a language modeling technique that helping machine to understand natural language. CKB stores structured data connected with relations, and the generated system can use it to generate paragraphs. In this chapter, techniques regarding NLG will be introduced.

2.1 Knowledge Base

Plain text is easy to read for human. However, machines can hardly understand the unstructured plain texts. KB encodes the real-world knowledge into machine-readable formalized representation. It's a domain-specific knowledge extracted from texts or constructed by knowledge engineers. It consists of basic rules and facts, including definitions, theorems and algorithms. In contrast to database which is composed of tables with flat data, strings and numbers, data in KB

linked in some relations. Data stored structured and categorized so that it can be accessed efficiently. KB could be applied in expert system (ES), which includes domain-specific KB and inference engine, to assist complicated problems that humans may make mistakes. E.g., a medical diagnosis ES are designed to assist doctors diagnose symptoms.

2.1.1 Commonsense Knowledge Base

Common Sense

Common sense is different from common knowledge. They are always mixed up. In fact, common sense is not sometimes really common to everyone in the world. It maybe only a group of people in specific time, location and background who would generally possess the knowledge. The commonsense knowledge includes daily life experience which nearly all people in the same community could infer imperceptibly without any prior or further learning. People use common sense to solve the problems they face everyday. It may cover a variety of domains, including social, physical, psychological, temporal and spatial aspects.

For instance, when a normal person see a puddle in raining day, he or she wouldn't step into it. Electricity is dangerous. Open mouth when eating apple. These are all basic knowledge that people don't have to learn on purpose. Common sense differs from person to person. The common sense of people in least developed country and in developed country is different because of their growing environment. Even if people in the same country, they may have different commonsense knowledge because of different experiences. Common knowledge is the known basic facts Some common knowledge reflects constant facts that won't be

changed as time goes by, e.g., the sun rises in the east and sets in the west.

Commonsense Knowledge Base

Some commonsense knowledge that are domain-specific and can be formulated as CKBs. They are developed to assist people solving problems in narrow applications. They would know the hallmarks of cancer or how to prevent cancer, but they don't know a basketball is round or square shape.

Commonsense knowledge base is a knowledge base which stores common sense and its relations. The coverage of common sense is much broader than other specific domains. It contains almost everything in our daily life, but the knowledge in it is shallower than other KBs. Each entity has its own attributes, e.g., the attributes of cancer are growing fast, gene mutation or no cure. Relations between entities, e.g., medicine capable of curing insomnia or see a doctor when having a sore throat.

Cyc Cyc [12] is a project started in 1984. It attempted to build a large formalized KB that assemble millions of human common sense and ontology. It contains about 2.5 million assertions, which consist of facts and rules, and over 1.5 million terms, including 42,500 predicates. It spent approximately 1000 person-years to capture a variety of domains common sense, and now is still developing.

WordNet WordNet[1] [13] is a semantic network with hierarchical structure which contains word semantic and ambiguous words. There are relations between words, including synonym, hypernym, hyponym, meronym and holonym. It emphasizes

[1] http://wordnetweb.princeton.edu/perl/webwn

lexical categorisation, and Cyc emphasizes logical reasoning.

2.1.2 Knowledge Acquisition and Representation

Knowledge Acquisition

Knowledge acquisition is the process of extracting meaningful information from text or people, and to be used in downstream tasks. There are roughly three methods to extract common sense. Acquiring data from plain text [14, 15], experts [12, 13] and crowd-sourcing [5, 16]. The first one is the quickest and cheap, but it may contain reporting bias [17]. The second one is time-consuming and expensive but having higher quality. The last one is more balanced compared to the first two.

Text

There is a tremendous amount of text on the Internet, such as social media, Internet forum or e-book. It would be helpful if it can be encoded into useful knowledge that machine can understand. Most of the knowledge captured is explicit knowledge, which can be expressed, understood and propagated easily by other people, such as text, number and image. It's difficult to acquire implicit commonsense knowledge directly from text [17] automatically. One of the difficulties is reporting bias [18], which refers to the frequency of events or actions in text do not necessarily match with the real situation in real-world. Here are some difficulties in attempt to extract relations from plain texts.

- Text on Internet is normally unstructured plain text. There is no fixed length in a sentence, no explicit structure and no rules. Full of redundant and

meaningless data. Lots of new words invented from social media everyday. It's hard to extract clean and structured information from plain text.

- People tend not to provide information that is obvious. For example, when someone says "The person walks in the park". He wouldn't add information about the person walks on foot. It's obvious to everyone that people walk on foot. Therefore, it's difficult to identify which implicit common sense from text or utterance is interesting and useful in generating text.

- People would share exceptional things on Internet rather than normal one. E.g., people are unlikely to discuss the experience of not seeing paranormal activities, they tend to share or discuss the information that is observed with others. Extracted information would indicate that people see paranormal activities is more common than not. Therefore, the importance of a relation between concepts discussed on Internet may not reflect the same as those considered in real-world.

- Common concepts in different sources would not be the same. Concepts expressed in newspapers or books are unlike concepts in Internet forum or social media. More specifically, the former tends to use more formal words, while the latter tend to use more informal and slang words.

- The topics discussed or described on sport news and travel news are somewhat different domain concepts. Sport news reports the things happened in sports especially for competitive sports. The words like win, lose, trade or retire are common in sport news. Travel news reports travel guidance so

that people can arrange their ideal itinerary. The words like bus, attraction, room and food are common in travel news.

Text on Internet is often heavily biased toward certain specific domains. Therefore, it's sometimes hard to acquire purely the commonsense knowledge from texts on Internet. It's also unstructured and its concept appearance frequency of words on Internet does not necessarily represent its importance in the real-world.

Experts

Some of the KBs constructed by experts, such as Cyc and WordNet. Knowledge engineers are intermediaries who encode high-level expertise from experts to machine-usable organized representation. There are several approaches to extract knowledge from experts, the following is the typical process of knowledge elicitation process.

- Knowledge engineers prepare some questions and interviewing experts directly.
- Experts answer designed questionnaires. Knowledge engineers collect and organize them systematically.
- Experts explain how or why they are doing during their problem solving.
- Knowledge engineers solicit the expert's problem solving by asking experts. Using this approach when there are still situations that experts can't clearly explain why or how they solve problems.

It's time-consuming, inefficient and expensive to build such a large KB using these approaches. It may spend over a period of years to complete. Experts can't cover a variety of domains due to the limit of human resources, although it may

contain fewer errors. The reliability of experts who participated the task also need to be identified. This approach is more suitable for the domain-specific KB. It needs expertise and less amount of knowledge compared to CKB.

Crowd-sourcing

Crowdsourcers distribute designed questions to unknown crowds. The types of crowds in building CKB are usually web users. There is usually no restriction about the participants as long as they can response the questions. Web users are willing to spend a short time when available to complete micro-tasks. Participants may get rewards (profits or reputation) or they may be entertained in the process of the micro-tasks if in a form of games. In contrast to experts, it's quick, cheap and more wide range of domains are covered. It is also more reliable than acquiring data from text. However, it still exists some problems which will be discussed in chapter 3.

Knowledge Representation

Knowledge representation means how data stored in KB after extracting from a large scale of text or from human crowd-sourcing. Structured and formalized data can be accessed efficiently and understood so that users can apply knowledge easily.

Semantic Network

Semantic network is a graph which contains nodes and edges. Each node represents a physical object or a virtual concept. Two nodes in a graph can be connected by certain kind of relation either with directed or undirected edges.

There are 6 different types of structures in semantic network [19]. Definitional networks is one of the most used structure. Relations between nodes are subtype or supertype relation. It defines type B is a subtype of supertype A. For example, dog is-a animal. The dog is a subtype of animal. Besides definitional networks, relation between nodes can be anything depending on different usages. Figure 2.1[2] shows relation between nodes can be "have" or "property".

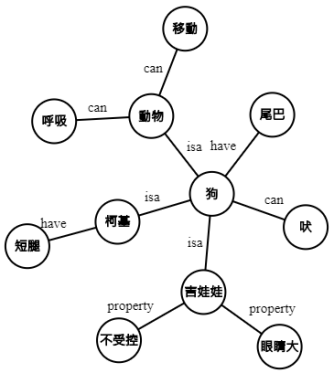


Figure 2.1: Relations between nodes in semantic network.

2.1.3 ConceptNet

ConceptNet [5] originated from Open Mind Common Sense(OMCS) [20] Project and developed by MIT Media Laboratory. It collected commonsense knowledge from voluntary web users all around the world, which are comprised of different kinds of persons.

It is a large-scale CKB based on crowd-sourcing and supporting 68 common languages and 10 core languages. It contains a variety of domains in real-world and can be applied in different tasks, such as analogy, commonsense reasoning and natural language understanding. Retrieve data by querying the provided web API[3] or downloading from official website[4] to a local computer. It consists of over

[2] https://csacademy.com/app/graph_editor/
[3] https://github.com/commonsense/conceptnet5/wiki/API
[4] http://conceptnet.io/

5,500,000 assertions (non-English languages are excluded) in English version, and over 580,000 assertions (non-Chinese languages are excluded) in Chinese version. Lots of data in ConceptNet is cross-language, such as "扣球 RelatedTo smash". We don't consider relations between Chinese and other languages in this research, though they contain over 560,000 assertions in Chinese ConceptNet.

ConceptNet is composed of several fields, such as concept URIs, source URIs, contributor, and so on. We only use **Start**, **End**, **Relation**, **SurfaceText** and **Weight** fields. An assertion consists of Start, End and Relation (Figure 2.2). **Start** and **End** fields are the first and the second concept of the assertion. They have fixed order that can't be swapped. A concept may be a word, words or a phrase (combination of words). Word is a basic unit that can represent semantic or pragmatic, and can be used independently. Concept in ConceptNet could be physical objects like television, tree, human or abstract ideas like anxiety, love or fear. **Relation** is syntactic or semantic connection between Start and End concept. There are different kinds of relations, such as AtLocation, IsA, HasSubevent, etc. Complete relations of English ConceptNet see appendix table A.1. **SurfaceText** field is the original natural language text that expressed statement. **Weight** field denotes the strength of the assertion.

Chinese ConceptNet

Chinese ConceptNet [21] collected data from community-based game. Methods like this called game with a purpose (GWAP) [22]. GWAP is more entertained and interesting than formal questionnaire survey, which may have more participants are willing to answer questions to achieve some goals in games. Furthermore,

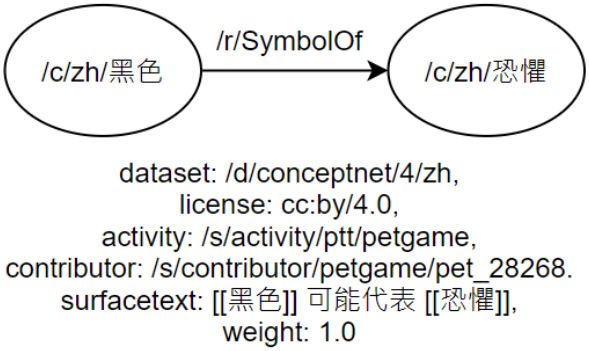


Figure 2.2: ConceptNet assertion.

it reduces the cost because participants are voluntary. Kuo et al. used Rapport Game on Facebook and the Virtual Pet Game on PTT (Taiwanese BBS). Both of them use question-answering template, e.g., "你會 B 因為你 A" (you B because A). Players can get rewards by filling Q&A templates. They can know their friends well in Rapport game, and players can teach pets common sense to make them smarter. Although ConceptNet is a so-called CKB, players fill in both common sense and common knowledge.

Each relation has corresponding parts-of-speech and roles. A relation may contain different meanings because players fill in slots according to SurfaceText not relations. SurfaceText with similar meanings grouped into the same relation. Detailed information and examples of each Chinese ConceptNet relation see appendix table A.2 and A.3.

The information of English and Chinese ConceptNet and relations with semantic meaning used in daily life are shown in table 2.1 and 2.2. Excluded relations in English version see appendix table A.1. Some relations are used in different purposes. They should be excluded when generating paragraphs because they don't have much semantic meaning which can be used in daily life. For example, rela-

tion “ExternalURL” is a link to external website which can describe that concept, and relation “DerivedFrom” means A is a word within B, e.g., fun → funny. Chinese ConceptNet data reduce 40% after removing non-daily life concept, and even 92% of data removed in English ConceptNet. Actually, there is not much data as imagined. In the following chapters, ConceptNet refers to Chinese ConceptNet, language of ConceptNet will be specified if necessary.

CKB	Assertions	Concepts	Relations
ConceptNet_en	5,577,656	1,557,614	40
ConceptNet_zh	586,595	328,577	26
	Used in daily life		
	Assertions	Concepts	Relations
ConceptNet_en	458,603	283,990	28
ConceptNet_zh	350,429	121,527	16

Table 2.1: English and Chinese ConceptNet information.

Chinese ConceptNet relations used in daily life			
AtLocation	CapableOf	Causes	CausesDesire
Desires	HasA	HasFirstSubevent	HasProperty
HasSubevent	IsA	MadeOf	MotivatedByGoal
NotDesires	PartOf	SymbolOf	UsedFor
Chinese ConceptNet relations not be used in daily life			
Antonym	DerivedFrom	DistinctFrom	EtymologicallyDerivedFrom
EtymologicallyRelatedTo	ExternalURL	FormOf	RelatedTo
SimilarTo	Synonym		

Table 2.2: Chinese ConceptNet relations.

2.2 Concept Association

One way to guarantee the coherence of generated text is using templates that capture the concept association among concepts. If concepts in a paragraph are associated, it would be more coherent. Concept association includes similarity

and relatedness (similarity is a subset of relatedness). Concepts are similar when they are in similar categories, and concepts are related when their semantics are associated in terms of similarity or relatedness. We need to have a method to infer the concept association extracted from the CKB. We will introduce some methods of evaluating association in this section.

Concept Similarity

Shortest path

$SIM_{Rada}(A, B)$: minimum number of edges (IsA) separating A and B in taxonomic hierarchical semantic network [23]. SIM_{Rada}(組織, 群體) = SIM_{Rada}(銀行, 央行). Their distance are both one step, but the relation between (銀行, 央行) is much more close than (組織, 群體). It doesn't take IsA depth into account. Similarity of concepts in lower layer is more similar than upper layer.

Least Common Subsumer

LCS means the first common ancestor from leaf to root node of A and B

$$SIM_{WU}(A, B) = \frac{2 * Depth(LCS(A, B))}{Depth(A) + Depth(B) + 2 * Depth(LCS(A, B))} \quad (2.1)$$

[24]. It takes the depth of concept into account. The similarity of (A,B) is directly proportional to $Depth(LCS(A, B))$.

Information Content-based

The concepts are more similar if they share more common information.

$$IC(concept) = -\log p(concept), p(concept) = \frac{tf + if}{N} \quad (2.2)$$

tf:term frequency, if:inherited frequency(sum of children's tf).

Resnik combines LCS and IC. $SIM_{Resnik}(A, B) = IC(LCS(A, B))$ [25]. The problem is any two pairs of concepts which have the same LCS will have the same similarity, e.g., $SIM(\text{動物}, \text{植物}) = SIM(\text{柴犬}, \text{西瓜})$. They both have the same LCS 生物.

Lin Add information about concepts in addition to commonality [26].

$$\frac{2 * IC(LCS(A, B))}{IC(A) + IC(B)} \tag{2.3}$$

Concept Relatedness

Path-based

Graeme Hirst proposed a method based on WordNet and defined three kinds of relations [27]. Concepts are extra-strong relation if they are the same word. Concepts are strong relation if they have parent in common or a horizontal link between them. Concepts are medium-strong relation if their linking pattern is one of the allowable paths in Figure 2.3. Each edge represents one or more steps from a concept to another one. It can only change the direction once to avoid a large semantic step.

Figure 2.3: Allowable paths in medium-strong relations.

Search engine based

Using page counts to measure the association, but it has several limitations. First, the returned page counts may be inaccurate. Secondly, two words are less relevant when the distance between them becomes too far away.

Thirdly, the result is incorrect when words have ambiguity meanings. Lastly, it can't access the search engine too frequently, otherwise the performance will become very inefficient.

Vector-based

Words and concepts are represented in terms of numeric vectors. It can compute and establish from some large scale of unlabeled plain text corpora which is efficient than the limited size corpus annotated or labeled by experts. It's intuitive that one can represent each word using one-hot encoding method. Mapping each word to an one-hot vector, it will become very high-dimensional and sparse, and the vector size is directly proportional to vocabulary size. This approach can be very large and hard to scale up because words are independent to each other, and don't have much mutual information.

Distributed representation represents words as low-dimensional and dense vectors. It's easy to scale up and the size is independent of vocabulary size. Distributional representation is based on distributional hypothesis, which is words have similar context in the encoding space tend to have the similar meanings. Words in distributional vectors are dependent and have more mutual information than one-hot encoding.

There are two approaches to construct word embedding, count-based (or frequency-based) and prediction-based. Count-based word embedding is the first-order co-occurrence of the words. It calculates syntagmatic relation between words by co-occurrence frequency within a context window size. Construct a sparse two-dimensional co-occurrence matrix, and using

Singular Value Decomposition (SVD) to reduce the matrix.

Prediction-based became more popular after skip-gram model [28] were proposed. It predicts surrounding context words within a context window size of each target word. Maximize the prediction probability of the given target word from the context or sequence words in the one-hot encoding representation.

Skip-gram model is a static word embedding method which doesn't consider the word ambiguity. The target word in contextualized word embeddings [29–31] has different vectors according to different context words.

Context window can be fixed or dynamic size. Fixed window is simple and quick, but its performance may affected by the meaningless words. Size 5 is normally used in many researches. It still depends on different tasks and corpora. Dynamic window relies on dependency parsing over the context words by syntactic [32] or semantic relations [33].

Word embedding can be used as embedding layer in neural network. It can also calculate relatedness between concepts when generating paragraphs. There are different types of distance metric, Jaccard, Dice, cosine similarity, etc. Most common metric in word embedding is cosine similarity: $\frac{A \cdot B}{\|A\|\|B\|}$. It calculates angle between two vectors. If they have same orientation, they may relate to each other and the value is close to 1. The range is between -1 and 1.

2.3 Discourse

Discourse means sets of connected sentences with some relations. It's natural and interactive. There is an agent or agents in the discourse. It can be either written, paragraph, or verbal form, dialogue or conversation. Discourse makes sense when it contains cohesion and coherence.

Cohesion refers to external text connectivity. Coherence refers to continuity of senses, it's internal relations of a sentence or discourse which is semantically logical and consistent so that it can be understood easily. Discourse contains related text is not necessarily coherent. Text has lexical links, but it doesn't share similar topics. Likewise, discourse contains coherent logics is not necessarily cohesive. Even though sentences have similar topics, they are not connected. Therefore, discourse makes sense when having both cohesion and coherence, they are independent and indispensable.

Discourse Cohesion

There are five types of cohesive ties (relations in text) based on Halliday [34]. They can be grouped into grammatical and lexical cohesion. Grammatical cohesion includes reference, substitution, ellipsis and conjunction. Reference refers to words don't have semantic meaning, it makes reference back to other words, such as I, this or there. Substitution refers to a word replaced by another word to avoid repetition. Ellipse is simplification of a sentence. Some words are omitted, and people can still understand it. It's a kind of substitution, and can be seen as replaced by blank. Conjunction binds words or sentences together in some relations,

such as temporal, causal or adding information. Lexical cohesion includes reiteration and collocation. Reiteration is the restatement of the same words. It can be repetition, synonyms, plesionyms, hypernyms or general words. Collocation is words that regularly co-occur.

Discourse Coherence

A discourse is semantically logical and consistent if senses are continuous. It can be used in NLG [35, 36], document summarization [37–39] and automated essay scoring [40, 41]. A discourse with high coherence is well-formed, easy to understand and strong words connection. A discourse with low coherence is unorganized, doesn't have a core topic and words lack connections. For example, if concept nodes are persons, the edge can be the friendship between nodes in a network graph. If there is a cluster that persons know each other, they may have same interests, such as they all like watching baseball games or hiking. Members in a cluster like that may have similar topics. In the same way, a coherent discourse is composed of related concepts that share some similar properties.

There are several types of coherence relations [42], occasion, evaluation, explanation, parallel, elaboration and contrast. Occasion relation means former sentences set up the occasion for latter. Evaluation relation is latter sentences indicate why former sentences to be said. Explanation gives the cause or result about former sentences. For example, Ming didn't take a bath today. The water got cut off. Parallel refers to sentences sharing similar properties. Elaboration refers to the elaborated sentences that describe the same thing but has more detailed than the other. Contrast relation is that sentences have the opposite meanings.

A coherent discourse has these relations, but a discourse that has these relations may not necessarily coherent. Take explanation relation as an example,

Ming was scolded by his boss. He were thinking about something.

It's hard to relate the concept sentences of thinking about something with the one being scolded by his boss. The complete explanations of this might be:

Ming were thinking about something. He forgot to put oil in the car. He took the bus instead. Because the bus was late, he arrived late for work. He was scolded by his boss.

There are several reasons to cause the final result. If the explanations are too many, the final result is less relevant to the first explanation. Therefore, it's difficult to evaluate whether a discourse is coherent or not.

2.3.1 Discourse Coherence Evaluation

A generated paragraph may consist of one or more cohesive ties and coherence relations in section 2.3. It's important to evaluate whether these relations exist in the generated text or not. There are two evaluation metrics, which are human rating and automatic evaluation, to evaluate the discourse coherence.

Human Rating

Human rating is the most natural and intuitive way to evaluate a discourse. It includes intrinsic and extrinsic evaluations. Intrinsic evaluation asks participants to rate a generated discourse based on coherence, content, organization, writing style and correctness [43], or performing a Turing test based questions to distinguish

between machine-generated and human-generated discourse. Extrinsic evaluation is a task-based [44] evaluation. Design a downstream task for users and evaluate the result. It usually has higher cost than intrinsic evaluation. Intrinsic evaluation isn't always better than extrinsic evaluation [45, 46], it depends on different tasks.

Automatic Evaluation

Because of the high cost of human rating, many automatic evaluation have been researched. Sentence ordering ranking is a simple method to evaluate coherence [47, 48]. Assuming original sentences is well-formed and coherent than any other random reordering. Compare original sentences with random reordering, the one close to original sentences has higher ranking.

Compare to human-written corpus Bangalore compared generated text and human-written corpus by the number of insertions, deletions and substitutions. [49]. Langkilde parsed the text from corpus and entered the parsed result to generated system. Compare the generated text to the one in the original corpus [50]. This metric is cheap, quick and repeatable. The quality depends on human-written corpus text. Belz and Reiter have shown that similarity to corpus evaluations tend to favour the repetitive texts than the variety ones [51]. It has lower correlation with the subjective human judgement than other evaluation methods.

Neural Network Based Map sentences to embedding vectors to capture semantic and syntactic relations. Unlabeled human-written text as positive samples, and random sentence reordering [52] or replacement [53] as negative samples. The language model is a model that can translate plain texts into a machine-readable

form, it can map unlabeled words to some low-dimensional and dense vectors to get word embeddings. A discourse consists of sentences and words. A simple sentence embedding is summing up all word embeddings and divide by the number of words to get average sentence embedding. In the same way, average sentence embeddings to get discourse embedding. Simple methods like summing and concatenating are also used commonly. These methods don't consider the word ordering but is very quick. There are also other methods to embed word sequence based on neural network [31, 54–58].

Automatic evaluation is cheap, quick and repeatable than human rating [59]. It is a supplement to human evaluation, not a replacement for it. Even though the processes of human rating are time-consuming and expensive to design, apply for participants, collect and organize data, it remains gold standard evaluation of machine-generated discourse. It's still a common metric in present researches.

2.4 Related Work

[60] generated a modern style Chinese poems based on templates and evolutionary computation which is similar to our research. The difference is they have to control the oblique tones for each lexicon, and the number of words in a sentence is fixed. The evaluation lacks global coherence, they only considered rhyme, tone, antithesis. These features can only ensure the correctness of grammar, they can't evaluate the text coherence. Another essay generation proposed by [61], they generated essay given some topics. Using a multi-topic-aware LSTM model to train the data and generate the essay. The downside is that they need a training

data which consists of topics and corresponding sentence. It needs much effort to annotate those data manually.

2.4.1 Human-Crafted Template-Based Generation

Template-based generation approach can usually ensure the quality of grammar. Some simple systems which don't have many variations are relatively easy to design, such as simple basketball game reports. Fill in the slots with player name, scores and number of rebounds, e.g., __ scored __ points and __ rebounds.

Realizers like FUF/SURGE [62], AlethGen [63] and RealPro [64] used a syntactic structure, e.g., tree structure with syntactic labeled nodes and arcs, to store input text and formatted linguistic KBs, which contain lexeme, part-of-speech (POS), morphology, etc., to assist syntactic, morphological and orthographic realisation, and generate grammatically correct sentences based on pre-designed templates.

SimpleNLG [65] provides a JAVA API[5] to generate grammatically correct English sentences given detailed information of subject, verb, object, and so on. Users specify the text and POS they need, SimpleNLG will decide text ordering, put whitespace and punctuations in the right place and make sure grammatically correct. SimpleNLG is similar to game reports system but with some post-processing not just filling in words.

Some templates need knowledge expertise to construct [66]. The process of designing discourse templates is time-consuming and hard to scale up because of the high-cost. In general, human-crafted templates based NLG lacks variation,

[5] https://github.com/simplenlg/simplenlg

adaptability and maintainability, but it ensures the quality of basic grammar.

2.4.2 Monte-Carlo Tree Search

The search space is too large to find optimal decision if each concept in CKB has lots of connective concepts. It becomes computationally intensive to exhaustive search each possible path from a root node to a leaf node. Monte-Carlo method was proposed by Stanislaw Ulam, it's a method based on random sampling to obtain numerical results. Remi Coulom combined Monte-Carlo sampling to the and game-tree search, and called it Monte-Carlo Tree Search (MCTS). MCTS is domain independent, it doesn't require any knowledge about the given domain to make decisions. In most MCTS-based NLG tasks, it plays a role in selecting words in a large space and evaluating the results.

One of the difficulties in template-based NLG is how to select the right words (traverse nodes) in such a large search space so that generated text is coherent. Mcintyre [67] indicated that the space of tree structure which is used to generate story can increase dramatically when the size of KB is large. The NLG experiments in [8] showed that MCTS outperformed other algorithms, such as BFS and DFS, in finding an optimal story either in a small or large search space. Therefore, random sampling in possible paths is better than exhaustive search.

MCTS builds an asymmetric tree by random sampling rather than symmetric tree. It makes MCTS possible to explore different decisions, which could avoid the same generated text in each iteration. It is not necessary to find an optimal decision in NLG tasks. A fixed output given fixed input is not what we expected because the generated text lacks variation even if it has strong coherence.

In [11], the author used MCTS to evaluate the quality of generated text. It generated text by reusing words from original text after some transformations, deletion or change of tense, and reordering the words. Evaluate the generated text by a reward function which consists of syntactical correctness and similarity to the original text. Train the samples from fictional movie conversations (original as positive and random words permutation as negative), to evaluate whether the text is syntactical valid or not. Compare the generated text to the original one by BLEU-4 [68] to get the score of similarity. Repeat the steps of random sampling and back propagation for a finite time and select the best generated text. But it only paraphrased user utterance without understanding the internal relations between words.

Jiwei Li proposed a model [9] based on generative adversarial network (GAN) [69] to generate and evaluate dialogues. Generative model is a SEQ2SEQ [70] model which concatenates two preceding utterances history as input and hopes to generate indistinguishable response sequences from human-written dialogues. Discriminative model distinguishes between human-written and machine-generated dialogues. The reward function of discriminator is approximated by one sample (a sentence). Each token (word) in a sample has the same reward (same positive reward or same negative reward). However, each token is supposed to have different rewards. They used MCTS to repeat the steps of random sampling in the rest of tokens given a partially decoded prefix sequence for N times. The reward of a specific token is the average score of random sampling of that token. But this model can't deal with problems in which a target utterance doesn't have context utterances. In [71], they indicated that text generated by SEQ2SEQ model lacks

of coherence, diversity and common sense.

Approach in [10, 72] is similar to ours. They use MCTS to select the words according to the context-free grammar (CFG) rules obtained from the Brown corpus. The classifier used original sentences in the corpus as positive samples, and random sampling as negative samples. Since it didn't have commonsense knowledge in the process of a generating sentence, they must relied on a large corpus to sample context sentences. We propose a model to generate natural language based on CKB without context sentences.

Chapter 3

Methodology

In this chapter, we describe the implementations of our proposed generated system. An overview of our system is shown in Figure 3.1. The user inputs an initial concept (subject) as the main topic which could be a living organism that can conduct the actions or concepts be described, such as person, animal or occupation, etc. ConceptNet consists of a large space of commonsense knowledge that is expressed in terms of various stereotype of triple-relations between concepts. We wish to extract the meaningful relations of concepts from ConceptNet to construct meaningful sentences and paragraphs as the basis of the text generation. We then adopt MCTS to simulate paragraphs in ConceptNet network and evaluate with a discourse coherence model. The aim of this research is to generate best-possible coherent paragraphs evaluated by our discourse coherence model rather than merely satisfying user's goal. User may or may not provide the initial concept. If the user doesn't specify an initial concept, the system can also generate an initial concept randomly to generate paragraph. And because the ConceptNet contains a large number of errors, we refined it and expanded the coverage of ConceptNet to improve the quality of generated paragraphs.

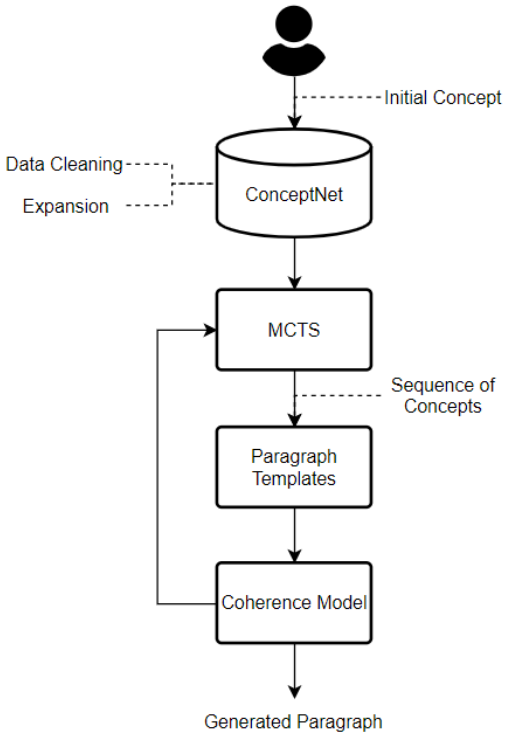


Figure 3.1: System overview.

3.1 ConceptNet Data Cleaning

Although ConceptNet is a large CKB, it contains many types of mistakes, such as typographical errors (typos), redundancy, relations in reverse order or meaningless data, etc. Input data is as important as model when training a neural network based system. Therefore, we refined ConceptNet to reduce its error rate. In the mean time, we increased the quality of ConceptNet. The quality here refers to correctness, coverage and number of concepts. Wider concept coverage isn't necessarily better, it may contain more errors because of crowd-sourcing limitations. We attempt to expand the coverage of each concept in ConceptNet and the number of concepts, and maintain its quality at the same time. We will describe different kinds of errors or downside in ConceptNet in section 3.1.1. In section 3.1.2, we will describe some methods that we adopt to clean data. In section 3.1.3,

we expanded ConceptNet by plesionyms (near-synonyms) in order to increase the coverage of ConceptNet.

3.1.1 Disadvantages of ConceptNet

Limitations of Crowd-sourcing

In order to reduce the cost of acquiring knowledge from experts or collecting manually, crowd-sourcing is an alternative method to acquire knowledge from the non-expert general public. It is more balanced compared to acquiring knowledge from experts or plain text corpus. The quality of translation datasets collected by the crowd-sourcing has been proved to be comparable to the professional translators given appropriate quality control [73]. English ConceptNet originated from OMCS, and Chinese ConceptNet used GWAP system to collect data. Both of them are crowd-sourcing projects. They collected common sense from voluntary web users all around the world.

However, the human crowd-sourcing techniques in collecting big data still exists some limitations. Semantic noise or errors of CKB in neural NLG is a problem which will affect the performance of the result [59]. The knowledge of expert systems is collected from domain experts. It is more specific, narrow and professional in some domains. Expert system KB contains in general fewer errors than the CKB collected from crowd-sourcing. Data Cleaning is required to replace or modify those incomplete, inconsistent and incorrect data in database which may lead to poor performance.

Knowledge acquired by crowds tend to be noisy, redundancy and meaningless especially for unguided projects without supervision and voluntary participants.

The coverage of uncurated crowd-sourcing may be wider than the curated ones, but the side effect is that it contains more errors. In order to increase the quality of answers, concise pre-statements are required. The situations of without supervision are similar to voluntary participants. The quality of data acquired from volunteer crowds may be lower than the paid crowds. This is because they usually answer the questions or fill in the answers based on their intuition. They don't ponder the questions carefully or pay much attention to it, and just want to finish the questions as soon as possible in spite of game-based questions. They usually ignore the pre-statements or instructions about that project. The pre-statements still can't guarantee the high quality of the results. Therefore, post-processing of evaluation and filtering is necessary after collecting the answers.

Missing Commonsense Links

Commonsense knowledge in real world actually is very large and versatile. It's hard to completely cover all. Therefore, lots of missing commonsense links between concepts are possible. For example, if the knowledge that Corgi is an animal not in the CKB, machine won't be able to infer that Corgi can breathe although it does know that animal can breathe. Without carefully carrying out the inheritance inference over the CKB can by itself commit errors. In [74–76], they discussed more about missing links in CKB.

Non-uniform Distribution

Figure 3.2 shows a non-uniform distribution of ConceptNet. The horizontal axis from left to right indicates the first N concept of frequency in descending order.

The red line represents the degree of concepts greater than 3 which may contribute to the paragraph generation (concepts with fewer connections to others may have lower contributions). The total degree of concept in the first 10% (12160) in ConceptNet accounts for 74.3% (517093/695854). The average degree is 5.7, and the first 10% is 42.5. Most of the data is in the range of the first 10%. Non-uniform distribution in ConceptNet will make generated paragraphs monotonous no matter how well the generated system is. Concepts need to connect with other concepts to generate paragraphs. The first concept connects to the second one, and the second one connects to next one, and so forth. If common concepts included in the path to generate, it will move back to the common concepts somehow after few steps. That or those common concepts will appear frequently in generated paragraph, and make it monotonous.

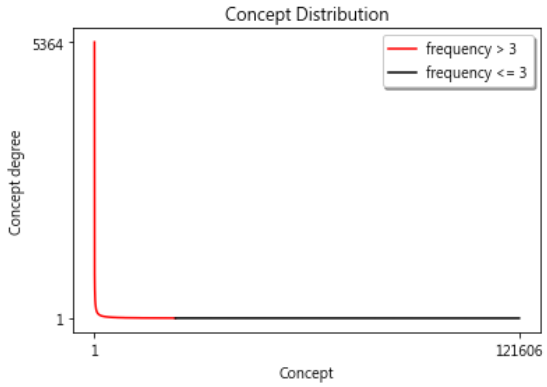


Figure 3.2: ConceptNet distribution.

Ambiguity of Concept

Common concepts tend to have ambiguity. An ambiguity in a sentence is a bit strange. For example, concept "squash player" relate to concept "play squash" and "play ball". If a sentence template is like "squash player ____ when play

squash/play ball". The blank can be "back to the T-position" or "hit the side wall" if related concept is "play squash". And the blank can be "slam dunk", "hit a home run" or "throw the ball" if related concept is "play ball". "play ball" is a more common concept than "play squash", but it contains ambiguities which are not suitable for this template. "play squash" is a rare concept compared to "play ball", but its connected concepts are more related to squash player. Therefore, higher degree of concepts is not always better than lower one.

3.1.2 Data Cleaning

Data cleaning methods (modification and deletion) which are adopted to clean data among all relations in ConceptNet will be listed in this subsection. Not all of the problems listed here are ConceptNet errors, some of them are modified to conform our system needs.

Modification

- Correct the typos by CKIP, a lab called Chinese Knowledge and Information Processing in Academia Sinica, Chinese Spelling Check System[1] [77]. Some typos have the same pronunciation, and some are similar words that usually be misunderstood, e.g., 綿、棉, 記、紀, 戴、帶. 戴 means wear something on body or face and 帶 means bring something. There are still lots of typos after automatic detection. The rest of them are corrected manually.

- Revert assertions which are in reverse order. A simple example is "[可食用] 是 [餅乾] 的"[2] ([edible] is [cookie]). Another example is "你可以在 [珠寶

[1] https://ckip.iis.sinica.edu.tw/service/typo/
[2] text inside square brackets [] refers to concept in ConceptNet

店] 找到 [珠寶]". The concept in Start or End field should be consistent. [珠寶店] (jewelry store) is a location which should be placed in End field of relation "AtLocation". However, the fields in ConceptNet is [珠寶店] AtLocation [珠寶]. The order in SurfaceText is not necessarily the same as the order in ConceptNet fields.

- Change the relations which are unreasonable or inappropriate between concepts.

- Concept in Start or End field in different relations has different parts-of-speech, such as verb in Start field of relation "CapableOf" is incorrect.

- Remove redundant words if they are already in SurfaceText, e.g., [上課的時候] 的時候，你會 [講悄悄話]。. Redundant words include 有點, 覺得, 時候, etc.

- Different SurfaceTexts in a relation have corresponding parts-of-speech and situations. For example, [公園] 的時候會 [慢跑] → 在 [公園] 會 [慢跑] ([jog] when you [the park] → [jog] when you in [the park]). The park is not an action that someone can take but a location. Another example is [鑽石] 的時候會想要 [咬] → [鑽石] 會令人想要 [咬].

- Remove subject like 你 (you) in SurfaceText to avoid specific pronoun. A generated paragraph has its own subject which is user's input. It's wrong to put subject in the SurfaceText again. It will conflict to original subject. Relations with pronoun in SurfaceText include AtLocation, Causes, CausesDesire, HasSubevent, MotivatedByGoal and UsedFor.

Concept Unification

Concept unification is a process of unifying similar concepts together. Unify Chinese variants which are words have the same pronunciation and semantic meaning but with different glyphs. Most of them are allographs (other writing), e.g., 濕、溼, 臺、台, 匯、滙. Unify segmented words in reverse order or have similar meaning, e.g., [很多作業] and [作業很多], [抹乳液] and [塗乳液] → [擦乳液]

Besides those unifications, ConceptNet is very non-uniform distributional. ConceptNet is incomplete due to limited resources and time. The number of concepts is imbalanced between common concepts and the others. The sum of first 10% concepts accounts for 74.3% in entire ConceptNet database. Balanced CKB is helpful when generating paragraphs. Decrease the number of common concepts to avoid paragraphs are always composed of them. It will make generated paragraphs more varied.

Because isolated assertions and low-degree concepts contribute less when generating paragraphs, they need to be decreased. Isolated assertions are totally isolated from the others in the semantic network. They can't be accessed by any other concepts. It would be a waste if they contain useful information. Low-degree concepts refer to concepts with fewer connections to others. Concepts with only one degree account for 65.3% (79398/121606) in ConceptNet. They have lower probability to generate paragraph successfully. Some potential nodes may not be simulated because the model spends time to explore low-degree concept which may be a dead end concept (can't be expanded any more). Most of them can't play an important role in paragraphs because they lack information to describe.

There are two methods to deal with low-degree problem. Remove these low-

degree concepts or increase their degree so that they are not low-degree. ConceptNet is large enough to include most of the daily life vocabularies, and isolated assertions or low-degree concepts are usually extensions or variants of existing concepts. Paraphrase these concepts to their most similar concepts (find similar concepts by average word embeddings) to decrease their number if they have. Increase low-degree concepts manually as more as possible if they don't have similar concepts to paraphrase. Each plesionym may have different connections with different concepts. Merge these plesionyms and share the information they have to increase the size of the group which have similar meanings. Decrease the number of low-degree concepts can construct a more complete concept with different descriptions, and avoid spending time to explore low-degree concepts. There are more choices when that concept is selected to generate paragraphs, and it will make paragraphs more varied.

Concept Abstraction

Concept abstraction is a process of abstracting core semantic of a concept, it is slightly different from concept unification. Remove prefix or suffix modifier is a simple example of concept abstraction, e.g., 很, 非常, 了, 的. Concept with detailed information make it more specific. KB is more rich and complicated if different kinds of individual characteristics concepts included in. However, constrained by the limited resources of KB. The more details of a concept, the less data it would be in ConceptNet. Therefore, we have to abstract specific concepts to its core concept, and removing detailed information to generalize. Concepts with specific number, name or location are unnecessary in KB sometimes, although

they provide more rich and detailed information. For example, [臺北市立民生國小] → [國小]. Taipei Minsheng Elementary School is still an elementary school no matter which city it located in. Another example, [全聯福利中心] → [超級市場]. Store name is also not important, though different supermarkets may have different products. Most of products in supermarket A could be found in supermarket B. Generally speaking, concept with or without those specific characteristics may not alter its core concept meaning.

Use ConceptNet relation "IsA" to find these specific concepts belong to which category. ConceptNet is a semantic network but not hierarchical structure like WordNet or E-HowNet[3] [78]. Select a random concept as a starting node, and find upper layer nodes by relation "IsA". Root node will diverge after few steps. Concept doesn't have an unique corresponding root node in ConceptNet. It won't become a formal hierarchical structure, even if concept has an unique root node. Because ConceptNet is constructed by voluntary web users, they fill anything which seem to reasonable in sentence template. They wouldn't consider overall structure. Figure 3.3 shows an example of root nodes of [學校] (school). Total number of root nodes is 1351. Some of them are related to [學校], but most of them are not. Web users fill concepts they can relate in template "[學校] 是一種 __" . It's less likely to build a hierarchical structure. Therefore, concept which is direct connected to starting node is better than root nodes for the use of abstraction.

[3] http://ehownet.iis.sinica.edu.tw/ehownet.php

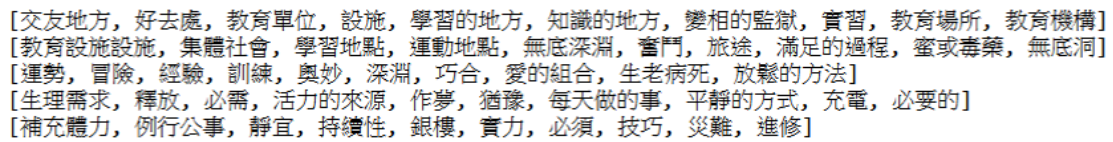


Figure 3.3: Root nodes of [學校].

Decrease the Number of Segmented Words

In order to translate concept to machine-readable representation, word embedding is needed. Because a concept may consist of words, it would be segmented to multiple words. However, average word embeddings of segmented words can't fully represent or even differ from original meaning, especially when the concept contains ambiguous segmented words or concept is a term that can't be separated.

We list two different levels of bias and how we deal with them. The first one is the concept meaning is completely different from the original one if it is separated (segmented).

Ambiguity

A word with fewer characters tend to have ambiguity, especially when a word is a single character. It may be an abbreviation of other words.

Take [賞 巴掌] (slap someone's face) as an example (blank between two words means they are segmented). "賞" means rewarding somebody for something, appreciating or admiring something. "賞" in [賞 巴掌] means giving something to someone. Word with ambiguities is hard to tell which one will be used in word embedding. Paraphrase [賞 巴掌] to its similar concept [打耳光] to disambiguate.

Metaphor

Some concepts can't represent their meaning if they are separated. They

imply a particular meaning that we can hardly know from each word's denotation. E.g., [腦袋 開花] → [腦袋 掛彩]. [腦袋 開花] doesn't mean flowers really blossom in someone's head. It means someone's head is injured. Idiom is similar to metaphor but with group of words having fixed combination and order, e.g., [輸到 脫褲子] → [輸光] (lose one's shirt at the track, English translation is not accurate.). Machine can't infer from [輸到] and [脫褲子] to [輸光] by averaging their word embeddings. A concept should be paraphrased to similar concepts if it is metaphor.

The second one is the concept meaning is biased from original semantic if it is separated though each word doesn't have ambiguity.

Segmented words combinations

Combine adjacent segmented words and rematch to vocabulary list to decrease the number of segmented words. For example, [聽 廣播] → [聽廣播] → [收聽廣播] (listen to the radio).

Remove prefix or suffix character of segmented word and combine with others to check whether it's in vocabulary list or not. For example, [蓋上 被子] → [蓋上被子](x) → [蓋上子](x) → [蓋被子] (tuck oneself in).

The meaning of combined segmented words is different from the original concept, even if the concept isn't metaphor and all of the segmented words don't have ambiguities. Concept semantic meaning is determined by its context. Combined segmented words and original concept usually don't have the same context, because they are separated when associating to context. Therefore, they don't have the same meaning as original concept.

For example, a segmented concept [返回 家鄉] (return to one's hometown) can be paraphrased to [返鄉]. Related concepts of [返回] (return) are [回到] (back), [離開] (leave), [前往] (go to), [抵達] (arrive). All of related concepts are about going to or back to somewhere. Related concepts of [家鄉] (a place that family has been living there for generations) are [故鄉] (someone's birthplace but no longer resident in there), [老家] (used to live in there), [南部](southern). Some of them are plesionyms of [家鄉] and some of them are about living in somewhere. All related concepts of [返回 家鄉] consist of related concepts of [返回] and [家鄉]. And related concepts of [返鄉] are [回老家] (return to the place someone used to live in), [探親] (visit relatives), [春節] (lunar New Year), [連假] (long weekend). It's obvious to know that combined of segmented words is different from the original concept.

Segmented words paraphrasing

The concept with more connections to others will make generated paragraph more varied. Paraphrase segmented words to short and common concept which are in vocabulary list to increase the connections to other concepts, e.g., [非常 好吃 的 食物] → [美食]. The less the segmented words, the more precise the word embedding can represent that concept.

In general, ConceptNet network after paraphrasing becomes more closely within a cluster than the original one, and word embeddings are also more precise.

Deletion

- Out-of-vocabulary (OOV), duplicate, some simplified Chinese concepts, offensive concepts, unreasonable and meaningless data, such as "[女人] 是一種 [錢包] ([woman] is a [purse])" or "[帥哥] 是為了 [生活]" ([handsome_guy] in order to [life]).
- Some data in ConceptNet are not precise, but not wrong technically. It could be rare conditions happen in special cases. However, these data are unhelpful when generating paragraphs. For example, it's unhelpful to know a table below universe, toilet paper is part of world or eating while jogging.

To summary this subsection, there are different levels of importance of modifications to make ConceptNet more suitable to generate paragraphs.

Number of connections between concepts The most important task is to generate paragraphs successfully, and it will be failed if there is not enough data to do it. The more connections between concepts, the more likely the paragraphs can be generated. If there is a region with concepts mutually connected in ConceptNet network, it tends to have similar topics or properties. A paragraph consists of similar topics would be more coherent.

Correctness of assertions and SurfaceText Incorrect concepts in a paragraph are obviously wrong and unreasonable. Different SurfaceTexts have different corresponding parts-of-speech.

Number of segmented words The semantic meaning of average word embeddings differs from original semantic when the concept is segmented. The

meaning will be severely biased when the number of segmented words increased.

Concept degree Provide more choices to generate more varied paragraphs.

Number of distinct concepts More different concepts can cover more different regions (topics) of ConceptNet network.

CKB size It's intuitive to know the quality of CKBs when they have considerable disparity in the size of database. Small CKB doesn't have enough data to generate paragraphs, even if the large one has much more errors than small one. It's hard to know which CKB is better when the size are closed.

Besides these conditions, the generated paragraphs need to be examined by human sometimes. Generally, size of these conditions are directly proportional to paragraph coherence and variation except for the number of distinct segmented words. The process of data cleaning in this subsection are among all relations, but each relation has different methods. The detailed information see appendix B.

3.1.3 ConceptNet Expansion

In order to make generated paragraphs more varied, we expand ConceptNet by different relations. User can also access data without precise concepts.

- Add new data to Start field of relation "CapableOf", "Desires", "HasA" and "NotDesires" manually. Increase the number of initial concepts, which is the root concept to simulate the MCTS, to avoid mismatching with user's input. Initial concept in those relations are usually a subject that can do actions.
- The paragraph will lack variation when the concepts are common and have

relatively less data in some relations compared to others. For example, the proportion of [考生] (examination candidates) in relation "NotDesires" and "HasProperty" is 126:1. There may be a template with "NotDesires" and "HasProperty" in the same time. If [考生] is the search node, a node connects with the other two nodes in a compound relation, of them. The generated paragraph may have different concepts in "NotDesires". However, no matter how many concepts in "NotDesires", there is always one fixed concept in "HasProperty". The bottleneck is controlled by the relation which has less data, even if the others have much more. Add new data to these concepts to balance the numbers in each relation.

Expanded by Antonyms

HasProperty

If A has property B, it may have property of antonym of B.

"[產品] 是 [昂貴] 的" → "[產品] 是 [便宜] 的".

[product] is [expensive] → [product] is [cheap].

Desires & NotDesires

If A doesn't desire B, A may desire antonym of B.

"[總統] 厭惡 [敗選]" → "[總統] 想要 [勝選]".

([president] doesn't want to [lose the election] → [president] want to [win the election])

If A desires B, A may not desire antonym of B.

"[女人] 想要 [減肥]" → "[女人] 懼怕 [發胖]".

([woman] want to [lose weight] → [woman] hate [gain weight])

Some of pairs are inappropriate.

“[人民] 懼怕 [旱災]” → “[人民] 喜歡 [水災]”.

([people] afraid [drought] → [people] like [flood])

Expanded by Synonyms

Synonyms are words which sound different but have the same meaning. In fact, most synonyms are plesionyms (near-synonyms) [79] rather than absolute synonyms which have exactly the same meaning. Absolute synonyms are fully inter-substitutable. They can be substituted for each other in any context situations without changing the original semantic meaning. They have the same denotation, connotation, implication, usage and POS. However, the pairs of absolute synonyms are rare. Therefore, we use plesionyms which are very close to absolute synonyms and mutually substitutable instead to expand ConceptNet.

Plesionyms usually don’t have the same meaning except denotation. Denotation and connotation are different aspects of a word. Denotation is a word’s explicit definition in dictionary. For example, denotation of table in cambridge dictionary is “a flat surface, usually supported by four legs, used for putting things on”. A word may have its connotative meaning not just denotation. Connotation of a word is its implicit meaning associate to emotion (positive, neutral or negative), past experience, environment or culture. A word can have different connotative meanings depending on different contextual situations.

Plesionyms have some central semantic features, i.e. semantic traits in [79], overlapping but differ in one or more peripheral features. For example, [苦痛, 苦

楚, 痛楚, 酸楚, 苦處, 苦水],[痛苦],[痛處],[苦頭, 苦難]. Although these plesionyms seem to be similar, they are not substitutable. Words in a sentence are closely related. Substitute words with plesionyms in a sentence or a paragraph will usually change its original meaning and style. The word would stick out like a sore thumb in a sentence if there exists slightly different in connotation. In order to find precise plesionyms that are very close to absolute synonyms, some conditions need to be excluded.

Ambiguity We exclude single character words and words which the number of categories in Cilin [4] more than one to disambiguate. We delete data of Cilin from 63,213 to 47,611. The goal is to increase the number of single category words as more as possible. Some of the ambiguities have relatively low frequency which can be ignored. If plesionyms consist of several groups (plesionyms in the same group are more similar than the others), we retain the group which is larger than the others.

Connotation

Formal/Informal Words in specific situations would be different. For example, [告訴] (tell) and [告知] (inform) both refer to say something to someone. [告訴] is informal and direct, and [告知] is more formal and usually used in business.

Archaic word Words are used in ancient times and still be used in book or movie, e.g., [爸爸] and [阿爹]. People barely used archaic words in everyday life. The meaning of some archaic words have been changed,

[4] HIT IR-Lab Tongyici Cilin (Extended), 哈工大信息检索研究室同义词词林扩展版

e.g., [大人] means father in ancient times, and now it means elders. Both of these two situations need to be excluded.

Emphasis Different plesionyms may have different emphases.

For example, [孤掌難鳴] and [無計可施]

[孤掌難鳴] emphasizes someone can do nothing by himself/herself.

[無計可施] emphasizes someone can't find any solution to solve the problem

Emotion [稚氣] (childlike) and [幼稚] (childish) both refer to an adult behave like a child. [稚氣] is a positive emotion to describe an adult is innocent, energetic, honest or curious like a child. [幼稚] carries a negative connotation which describes an adult behave badly like a child, such as stubborn, moody or immature. Emotion strength like [生氣](mad), [火冒三丈] (foam at the mouth) can be substituted though they are different. Different emotion strengths are acceptable.

Word frequency The word frequency < 130 are excluded.

Multiple segmented words The new concept replaced by plesionyms may not represent the original semantic meaning. Segmented words are closely related, e.g., [隱藏缺點] → [躲藏缺點]. Although [隱藏] and [躲藏] have similar meanings, they are not mutually substitutable.

Part-of-speech Exclude plesionyms which have different parts-of-speech. They can't be substituted for each other.

Besides excluding plesionyms from different sources, we also add new data manually. Find top 15 similar concepts by word embedding for each concept. Add

these data if they are in ConceptNet (avoid OOV) and not in synonym set, which consists of different sources. The total number of plesionyms is 16843, and 7774 match the concepts in ConceptNet will be used to expand.

3.2 Word Embedding

Training Corpora

We use 109 popular boards in PTT [5] and Chinese Wiki as our corpora.

Corpora	Words	Vocabulary size (frequency > 29)	Capacity
PTT,Wiki	858,118,783	249571	5.38 GB

Table 3.1: Word embedding training corpora.

Corpus Preprocessing

Remove tags, punctuations, numbers, non-Chinese, invalid and duplicate characters. There are some fixed format in different boards, such as opening hours of stores, the number of products or the reports written by which reporter. These words should be removed because they are not related to content. Because the fixed word length of one line (about 40 words), a sentence may be separated to multiple lines. Recombine these separated words back to original words if they are in vocabulary table.

Stop words removal Use stop words to remove some of content words and function words which have little semantic meaning. Content words include numerals and quantifiers. Numerals and quantifiers don't affect the main se-

[5] Taiwanese Bulletin Board System(BBS) which consists of 13,309 boards in plain text.

mantic meaning of the concepts even though they do have meanings. Although the number of objects has slightly different in context, they are still the same thing. It doesn't change any of its physical properties. For example, someone can drive a car, but can't drive two cars simultaneously. The main semantic meaning of a car or two cars are both cars. The number of cars doesn't change any property of it. It still has four tires, headlights, car doors, engines, and so on.

Some adverbs like adverbs of degree or time can be filtered out. Adverbs of degree are used to strengthen the meaning of adverbs or adjectives. Adverbs of time are used to describe when or how often an action happened. A concept with or without these words are basically the same. For example, this dress looks elegant and this dress looks very elegant. I buy a cake today and I bought a cake yesterday. It's not important how intensity an action, an adverb or an adjective. It's also not important when did you buy something. The point is how concepts interact to each other. Remove these types of adverbs may lose some information within sentence or paragraph, but the relations between concepts are still retained.

Some adverbs like adverbs of frequency can't be filtered out. For example, kids are always late for school and kids are seldom late for school. It's totally different how often kids late for school. It will change concept semantic meaning if they are removed.

Function words include conjunctions, particles, prepositions, interjection and onomatopoeia These function words have little concrete semantic

meaning and for grammar purposes such as 和、的、從、啊、砰. They are used to make sentence grammatically correct and relate to other words in grammatical relation. They don't change the relations between concepts. For example, the students study in a classroom. It doesn't matter which specific classroom they are studying in. The relation between students and classroom is still "AtLocation" when function word "a" is filtered out.

Function words may decrease the performance of word embeddings. If there are many function words in a sentence, it would become fragment after segmenting without removing them. The strength of relation between content words without removing function words may lower than the sentence segmented after removing. It's possible that most of the content words relate to function words because of their high frequency in paragraph. It's intuitive to know that collocated words have stronger relation than others. Function words may relate to each content word, and this make them meaningless in word embeddings. It's important to select stop words. Have to consider each situation whether stop words have semantic meaning or not.

Text segmentation We use Jieba[6] API to segment words. Although it can find unknown words, which are not in self-defined dictionary, by a HMM-based (Hidden Markov Model) Viterbi algorithm, it also finds lots of unreasonable words, e.g., 裡你寫, 講你領, 將你傷. The distinct words contain "你" are found 9598 times in the result of discovering new words, and the other one are found 126 times. Therefore, we don't use HMM to combine unknown words. Exclude word frequency < 30 to reduce the impact of unreasonable

[6] https://github.com/fxsjy/jieba

words and rare words. We recombine segmented words back to their original word, e.g., [喝 了 酒] → [喝] [酒] → [喝 酒]. “了” is a stop words [喝] and [酒] are separated because of removing a stop word “了”. Recombine these separated words so that it can represent its original meaning.

Count-based

Co-occurrence Matrix Build word-context co-occurrence matrix which is sparse and symmetric. In order to save the memory, we store the co-occurrence frequency only in upper triangular matrix. It saves half of the memory and speed up the computation.

The importance between target word and each context word is not the same. The closer to target word the more important it is. We adopt linear distance weighting in (3.1). Weight decreases as distance increases from target word.

$$[\ldots, target, 1, \frac{d-1}{d}, \frac{d-2}{d}, \ldots, \frac{1}{d}] \quad \text{d:distance to target word} \tag{3.1}$$

Weighted Co-occurrence Matrix Instead of using raw co-occurrence frequency, we use pointwise mutual information (PMI) [80] in (3.2) to weight.

$$PMI(x,y) = \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{NC(x,y)}{C(x), C(y)} \tag{3.2}$$

$p(x)$: occurrence probability of word x
$p(x,y)$: co-occurrence probability of word x and y
$C(x)$: occurrence of x, and N is total number of words in corpus
$C(x,y)$: co-occurrence of x and y
N : total number of words in corpus

It calculates how much more x and y co-occur. The range of PMI is from $-\infty$ to $min[-\log p(x), -\log p(y)]$. The value is 0 if x and y is independent (co-occur by chance) in corpora. The value is maximized when x and y are perfectly associated (always co-occur), and the value is minimized when x and y barely co-occur. Words with strong association have higher PMI value (high C(x,y)). PMI has different variants, such as positive PMI (PPMI) [81], PMI^k [82], Normalized PMI (NPMI) [83] and shifted PPMI(SPPMI) [84]. PPMI sets negative PMI value to 0. It makes sense to mark low correlation(tend not to co-occur) and uncorrelated (never co-occur) word pair to 0. The weighted matrix becomes more sparse since negative values are removed, and the computation cost is lower than PMI. PMI^k and NPMI were proposed to make PMI less sensitive to rare words. Levy and Goldberg [84] found that skip-gram with negative sampling (SGNS) is implicitly factorizing a shifted PMI co-occurrence matrix, hence SPPMI is the original PMI shifted by $\log k$ (k > 0).

$$SPPMI(x, y) = Max(PMI(x, y) - \log k, 0) \tag{3.3}$$

Dimensionality Reduction The size of our weighted matrix is 249571×249571 which is extremely large. We use truncated Singular Value Decomposition (truncated SVD) to reduce high-dimensional matrix. The formula of SVD and truncated SVD is in (3.4). $X_{m\times n}$ is a $m \times n$ co-occurrence matrix. $U_{m\times m}$ and $V^T_{n\times n}$ is left and right singular vectors. $\Sigma_{m\times n}$ is a diagonal matrix containing non-negative real number (singular values) on the diagonal in descending order. Truncated SVD discards values except first r largest singular values, they contain most of

information in original matrix. It ensures the minimal loss of information and dimensionality reduction as well.

$$X_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n} \rightarrow U_{m\times r}\Sigma_{r\times r}V^T_{r\times n} \tag{3.4}$$

Weighting Exponent Caron p-transform [85] adjusts the scale of singular values in diagonal matrix Σ by weighting exponent p, where $X = U\Sigma^p$

Principal Components Removal Another method similar to Caron p-transform is principal components removal (PC-removal) [86]. It removes the first k dimensions of Σ. High variance dimensions may contain more unuseful information to lexical semantic tasks in contrast to low variance dimensions. If mapping the first k dimensions back to the co-occurrence space, most contributing words are people's names, "and" and "or". Figure 3.4[7] shows the the performance of dimensions removal in different levels of random noise. The performance falls off slowly (some information lost) if removing dimensions from matrix with a small amount of noise. It shows significant effect of noise reduction if removing dimensions from matrix with large amount of noise. Optimal value of weighting exponent p and the number of removed first k dimensions depending on tasks and corpus.

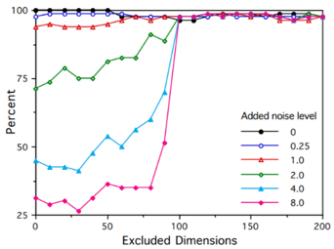


Figure 3.4: Dimensions removal.

[7] This figure is from [86]'s experiments.

Matrix Normalization Normalize the truncated SVD matrix to speed up the computation. Each row vector is unit vector (length = 1), therefore, the cosine similarity is dot product of two word embeddings.

After weighted co-occurrence matrix and dimensionality reduction, the count-based word embedding can be formulated in (3.5)

$$EM = [SVD_p(WCM)]_{m \times k:r} \tag{3.5}$$

EM : embedding matrix
WCM : weighted co-occurrence matrix
p : weighting exponent
m : number of rows (words)
k : remove first k dimensions
r : number of columns after truncated SVD

As to prediction-based word embedding, we use GENSIM [8] to train our models which are skip-gram and CBOW (Continuous Bag of Words). The detailed implementations are in their official website, and it won't be introduced here.

3.3 Discourse Coherence Model

3.3.1 Deep Neural Network Architecture

There is no clearly result of how coherent a paragraph is, it's hard to evaluate. We build a deep neural network (DNN) and attempt to evaluate the discourse coherence of paragraphs, this model is used as a reward function in MCTS. The

[8] https://radimrehurek.com/gensim/

architecture of our model is shown in Figure 3.5. A paragraph consists of sentences, and a sentence consists of words. The word embedding we build in section 3.2 are used as an embedding layer to encode plain text to word vectors. In addition, we need a word sequence encoder which encodes words to a sentence, and a sentence encoder which encodes sentences to a paragraph. Since the contribution of each word in a sentence is not equal, and each sentence doesn't contribute equally either. Also, the importance of the same words or sentences in different context are different. The relations between words or sentences and their context are highly dependent. Therefore, we includes two levels of attention mechanisms inspired by [87]. One in word level, and one in sentence level.

Word Level

Word Encoder First, we need to encode plain text $W_{il}, l \in [1, L]$ to word vectors x_{il} by an embedding layer W_e. Different words have different indexes in embedding matrix. Input sequence of word vectors to a BiLSTM (Bidirectional Long Short-Term Memory), and output the concatenation of forward and backward hidden states of BiLSTM.

$$\begin{aligned} & x_{il} = W_e w_{il}, l \in [1, L] \\ & \overrightarrow{h}_{il} = \overrightarrow{LSTM}(x_{il}), l \in [1, L] \\ & \overleftarrow{h}_{il} = \overleftarrow{LSTM}(x_{il}), l \in [L, 1] \end{aligned} \qquad \begin{aligned} & i : \text{ith sentence} \\ & l : \text{lth word} \end{aligned} \tag{3.6}$$

Word Attention Since not every word has the same importance in a sentence. The attention mechanism is introduced by [87]. u_{il} is the hidden representation of h_{il}. W_w and b_w is the weight and bias in word level. u_w is context vector in

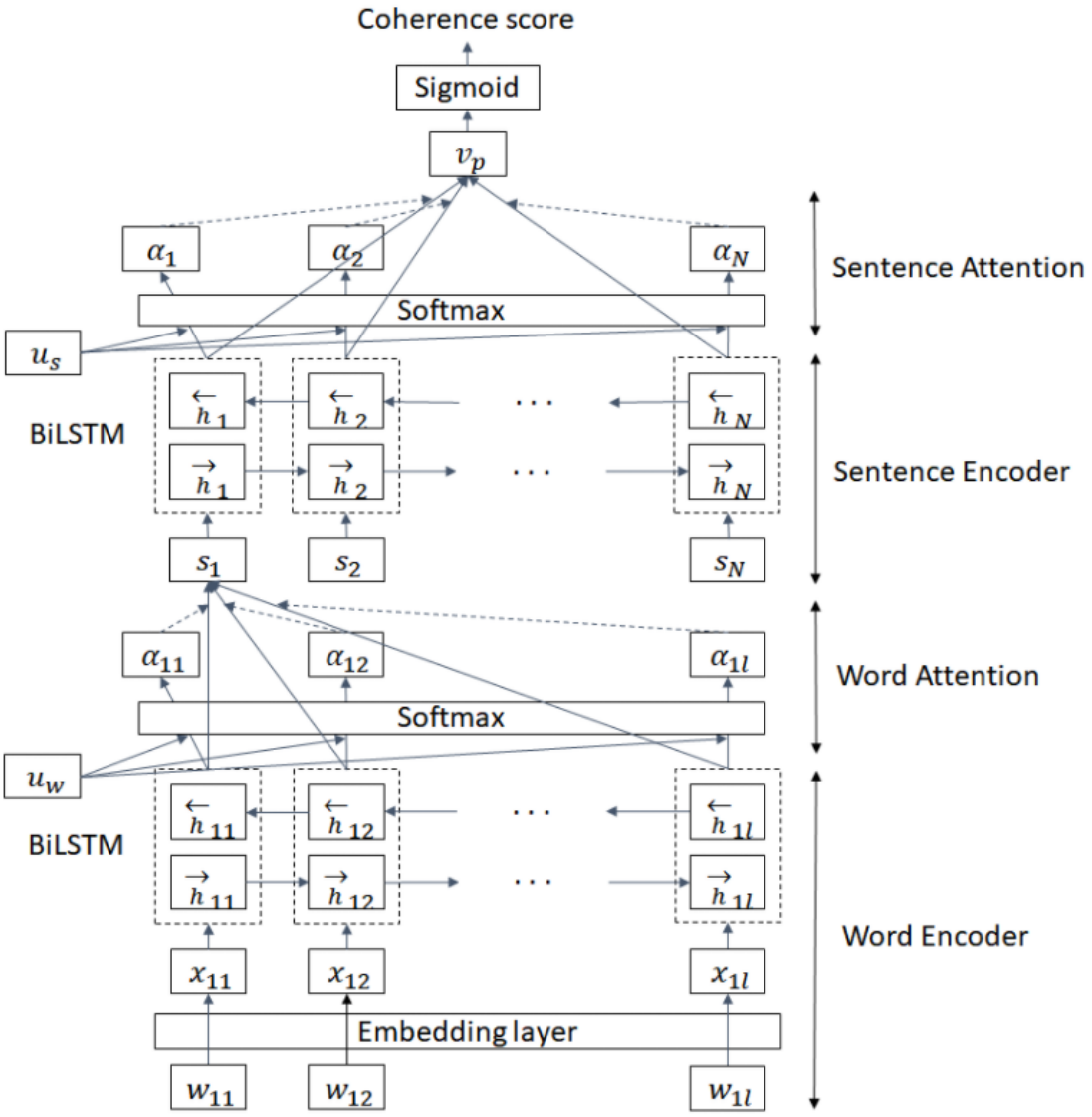


Figure 3.5: Discourse coherence model.

word level, it is randomly initialized and jointly trained. α_{il} is the normalized importance weight by softmax of the ith sentence and the lth word. Weighted sum importance weight α_{il} and the output of BiLSTM h_{il}

$$
\begin{aligned}
u_{il} &= tanh(W_w h_{il} + b_w) \\
\alpha_{il} &= \frac{exp(u_{il}^T u_w)}{\sum_t exp(u_{il}^T) u_w)} \\
s_i &= \sum_t \alpha_{il} h_{il}
\end{aligned}
\tag{3.7}
$$

Sentence Level

Sentence level is the same as word level. v_p is the vector representation of paragraph.

Sentence Encoder

$$\begin{aligned} \overrightarrow{h}_i &= \overrightarrow{LSTM}(s_i), i \in [1, N] \\ \overleftarrow{h}_i &= \overleftarrow{LSTM}(s_i), i \in [N, 1] \end{aligned} \tag{3.8}$$

Sentence Attention

$$\begin{aligned} u_i &= tanh(W_s h_i + b_s) \\ \alpha_i &= \frac{exp(u_i^T u_s)}{\sum_i exp(u_i^T) u_s)} \\ v_p &= \sum_i \alpha_i h_i \end{aligned} \tag{3.9}$$

Lastly, input paragraph vector v_p to a sigmoid activation function, and output a coherence score which in range 0 (incoherent) to 1 (coherent). We use layer normalization (LN) [88] to reduce Internal Covariate Shift (ICS). ICS is the covariate shift between hidden layers. The input of each layer is affected by previous layers, the following layers need to adapt new inputs. There are several methods to solve ICS problem.

- Smaller learning rate can limit the shift of distribution, but it spends longer training time.
- Proper parameters initialization, but it may spend more time to tune hyperparameters.
- Non-saturated activation function, e.g., ReLU.

- Batch normalization (BN) [89]. It calculates the mean (μ) and standard deviation (σ) of a mini-batch instead of a whole set, and $BN(z)$ is zero mean and unit variance. It prevents the training from getting stuck in the saturation regions (gradient close to 0), and lead to gradient vanishing.

$$BN(z) = \gamma \frac{z - \mu}{\sigma} + \beta \tag{3.10}$$

  z : activation function input
  γ : re-scale parameter
  β : re-shift parameter

- Layer normalization. BN is sensitive to mini-batch size. The performance of BN degrades when training on small batch size. The mean and variance are not accurate when the batch size is too small. Also, BN is difficult to apply to recurrent neural network (RNN) or LSTM. The length of sequence in RNN is not the same in different time steps. LN calculates mean and standard deviation of a sample instead of a mini-batch. There is no restriction with the batch size. LN has better performance in stacked RNN-like structure.

3.3.2 Training of the neural network

Corpora

The training corpora are shown in table 3.2. We use 71 popular boards in PTT. The total capacity is 4.82 GB after preprocessing.

	PTT	Chinatimes	LTN	Total
# of raw articles	5,794,074	2,696,580	957,146	9,447,800
# of segmented words	341,819,433	275,505,429	132,413,960	749,738,822
Capacity	6.79 GB	3.79 GB	1.78 GB	12.4 GB
Date	~2018/12	2009/10~2018/12	2005/1~2018/12	

Chinatimes: 中時新聞網
LTN: 自由時報

Table 3.2: Coherence model corpora.

Corpus Preprocessing

Most of the corpus preprocessings are the same with word embedding in section 3.2. Besides those preprocessings, the corpus in NN model is a bit different from word embedding.

- Exclude articles with fixed format in PTT, e.g., 交易、廣告、自介、團購、情報,etc. The sentences in these types of articles are listed one by one. They may be a name, address, date or site. Each item is not coherent at all.
- Exclude upvote and downvote replies. Each reply is short, it may be a word, a single sentence or the number of sentences < 3 because of the limited width in one line. Adjacent replies may or may not related depending on the user reply to article or other users' replies. We exclude replies to ensure the coherence of sentences. Most of the pages in Wiki are introduction of people, locations or events. They aren't suitable for positive samples.
- Period and exclamation point represent the end of a paragraph. Comma represents the end of a sentence. Therefore, we split articles to paragraphs according to period and exclamation point.
- Exclude the number of sentences < 5, it would not be a paragraph if the

number of sentences is too few. Exclude paragraphs which include single word sentences, they provide little information to the paragraphs. Exclude single words if they account for more than 30% in a paragraph, more than 75% in one sentence or the number of continuous single character words exceeds 2 in a sentence. Single word may have ambiguity or it may be a fragment of a word after segmentation or it is a OOV. No matter in which situation, it is better to remove. Exclude the number of OOV words > 3.

- Remove the last name before words, which are occupations or titles, e.g., [陳 醫生] →[醫生]. The last name in words are not important.

- Combine separated proper noun and people's name. Most of the proper nouns and people's name are not in the self-defined dictionary. The text segmentation API can't detect those words properly. Combine these words to reduce the number of OOV words.

Training Data

The coherence of paragraph is stronger than article. It contains fewer words and the strength of relations between words is stronger than articles. Paragraph is local coherence and article is global coherence. Therefore, we use paragraphs as our training data instead of articles. The positive samples are the original paragraphs, and the negative samples are paragraphs with replacement of other concepts. Words in a sentence or a paragraph are closely related. Concepts replacement makes paragraph less coherent even a single replacement.

Because we use MCTS to select concepts randomly and find the suboptimal

paths. If the replaced concept is less related to other concepts in the paragraph, the model can distinguish the incoherence easily. Therefore, we imitate the pattern that MCTS simulates randomly to get negative samples. We first select a target concept (a word or words) and check whether it exists in ConceptNet or not. If it doesn't exist in ConceptNet, it won't be used as a target concept. MCTS doesn't select concepts randomly with arbitrary concepts in vocabulary table, it selects concepts randomly in ConceptNet. Therefore, we replace adjacent concepts (context) of target concept by connected concepts in ConceptNet. The positions of adjacent concepts can be different, we have three modes to replace concepts. The first one is replacing two words, one before target and one after target. The second one is replacing two words, both of words are before or after target. The last one is replacing one word, before or after target.

The concepts used to replace the context are not arbitrary connected concepts. Not each connected concept can replace the context, they may have different parts-of-speech. The paragraph becomes very incoherent if it is replaced by arbitrary concepts, and it's not compatible to our MCTS-based model. Therefore, the POS of replacing concept must be the same as replaced concept. We use CKIP tagger [9] to tag POS. (There is an another old version of CKIP tagger [10].) The word segmentation we use is Jieba, and the POS tagging we use CKIP tagger. After comparing the result of word segmentation, CKIP tagger segments words to smaller unit which means more segmented words. For example, CKIP tagger:[一 大 堆 垃 圾] and Jieba:[一 大 堆 垃 圾] (use same self-defined dictionary). As mentioned in data cleaning subsection (3.1.2), average word embedding of segmented words

[9] https://github.com/ckiplab/ckiptagger
[10] http://ckipsvr.iis.sinica.edu.tw/

can't fully represent or even differ from original meaning. Jieba segments words to fewer number of segmented words which meet our needs. There is no good or bad here, it depends on users' tasks, and the CKIP performs better in POS tagging. However, we use different model of word segmentation and POS tagging. Some of the words segmented by Jieba are segmented by CKIP tagger again to tag POS. Consequently, we have to recombine the segmented words back to their original words, e.g., 航空 (Na) 公司 (Nc) → 航空公司 (Nc). Additionally, CKIP tagger divides POS into lots of classes. We don't have to get detailed POS, hence we simplify the POS in appendix D..

The paragraph templates can ensure grammatically correct. Although we don't define which template slot corresponds to which specific POS and the template slots may have different types of concepts, the POS in specific slot is basically the same. For example, a compound relation "CapableOf-HasSubevent" has a template "__在__會__", "人在公園會散步" (someone walks in the park). The first slot is location which is adverbs of place, and the second slot is an action which is a verb.

The replacing concepts can't be the synonyms of replaced concept. Otherwise, there is no difference after replacement. We find synonyms of replaced concept by word embedding (top 30 most similar concepts), Cilin and relation "Synonym" in ConceptNet.

The two examples of negative samples are shown in Figure 3.6. The text in red is target concept, and the text in blue is replaced (in upper) and replacing (in lower) concept which connected to target concept in ConceptNet. We can see that only a word or two words are replaced, the paragraph becomes incoherent.

Figure 3.6: Examples of negative samples.

In order to ensure the negative samples are truly replaced by other concepts. We iterate the replacing process until the number of replaced sentences > 1. We also tag processed concepts so that they won't be used in following iterations to reduce computational cost.

3.4 Paragraph Generation

3.4.1 Templates

We use sentence and paragraph templates to make text grammatically correct. We create 90 different combinations of relations in ConceptNet in appendix F. The combination of two relations which called compound relation. For example, the combination of AtLocation-HasProperty is "[公園] 的 [貓] 是 [黑色] 的". However, only few of them can be used in paragraph templates. The number of combinations in relation "MadeOf", "PartOf", "SymbolOf", "MayUse" and "HasA" are 39. But the common sense of these relations are difficult to be applied to paragraphs. Some of these templates are not possibly to be said in daily life or there are no appropriate relations can match with. For example, the compound relation of HasSubevent-MayUse is "[約會][喝咖啡] 時會用到 [杯子]". This sentence is semantically and grammatically correct, but it is less likely in our daily life conversations. Therefore, we only use 23 different compound relations in paragraph templates.

We create 8 different paragraph templates in appendix F. They are combinations of sentence templates. One of the example in table 3.3. The number of

CapableOf	MotivatedByGoal	
CausesDesire	HasProperty	
HasSubevent	MotivatedByGoal	HasProperty
HasProperty	Causes	
[老公][工作] 是為了 [有錢]		
[煩悶] 的 [工作] 會令人想要 [逃避]		
[有錢] 的時候會 [買房子]，是為了 [快樂] 的 [生活]		
但 [痛苦] 的 [房貸] 會帶來 [壓力]		

Table 3.3: Paragraph template

sentences in paragraphs is four or five, and each sentence has two or three relations. The first concept in paragraph template is a person, animal or something can conduct the actions (physical or mental) or be described. Some of the paragraph templates have backup sentence templates because concepts may not have corresponding connected concepts in some relations.

3.4.2 MCTS-based generated system

We combine coherence model and templates into our MCTS-based generated system. The flow chart of the generated system is in figure 3.7. User inputs an initial concept (subject), we first check whether it exists in ConceptNet or not. If not in ConceptNet, we substitute initial concept for its plesionyms or user can reinput it. And we adopt MCTS to simulate different combinations of concepts and evaluate coherence of generated paragraphs by the coherence model. We paraphrase the paragraphs generated by MCTS with plesionyms and pronouns. The replacement of pronouns can use Cilin (category A is human and category Bi is animal) or Named Entity Recognition (NER) to tag subjects as 他 or 牠. Evaluate those

paraphrased paragraphs by the coherence model again. Lastly, select one of the best results as our final generated paragraphs.

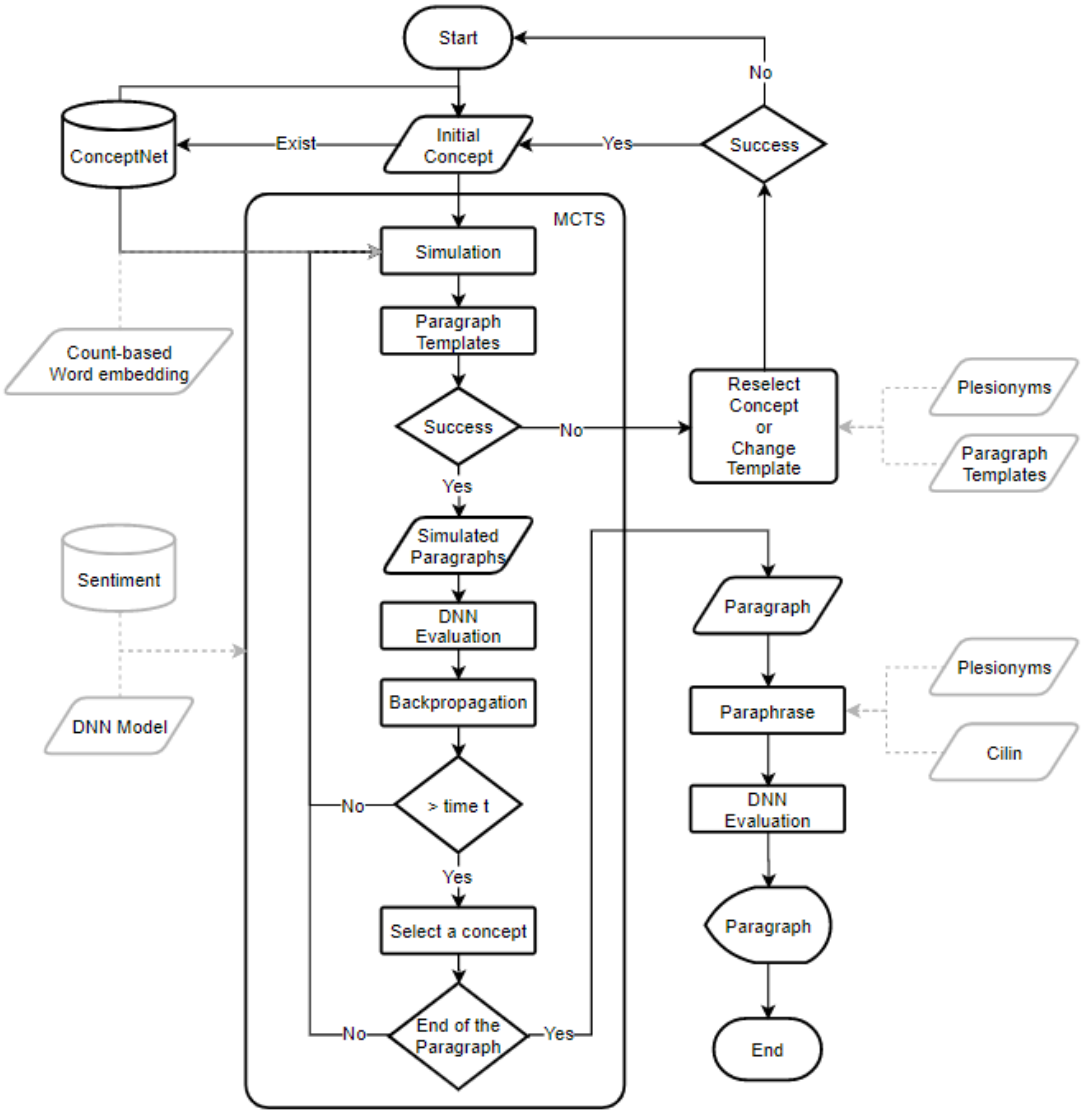


Figure 3.7: System flowchart.

MCTS

As we discussed in section 2.4.2, the search space is too large to find optimal or suboptimal paths by exhaustive search. MCTS is a heuristic algorithm which can find suboptimal paths in limited resources or time. An optimal decision algorithm is unnecessary in NLG tasks. Although it will find the most coherent paragraph,

it only finds single or few paragraphs as long as the initial concept has not been changed. We can view the problem of generating paragraphs as a decision tree search problem. MCTS builds an asymmetric tree instead of unfolding all possible paths. It can search coherent paragraphs in an extremely large tree. There are four steps in each round of MCTS which are selection, expansion, simulation, backpropagation. We will introduce these four steps in following descriptions.

Selection

Starting from a root node which is user's initial input concept. The tree continue traversing until reaching a leaf node. Each node in paths is concept that connected in ConceptNet. The selection in MCTS is a multi-armed bandit problem [42]. The gamblers have to evaluate the tradeoff between exploration and exploitation. They hope to maximize expected gain in limited resources. The simple selection strategy is ϵ-first. It is a pure random exploration in the first ϵT trials, followed by pure exploitation in the remaining times (1-ϵ)T. There are also other selection strategies, such as ϵ-greedy or ϵn-greedy. All of these strategies have some drawbacks. Either it has the same probability to choose good or bad arms in exploration phase, or the same expected gain arms have the same probability to be chose without considering the selected times in exploitation phase. Kocsis et al., [90] used Upper Confidence Bounds (UCB1) in MCTS called Upper Confidence bounds applied to Trees (UCT) as a tree policy in selection phase. In formula 3.11, $\frac{Q(v_i)}{N(v_i)}$ is the exploitation term. The child node v_i with higher score Q has higher probability to be selected. The tree should explore other nodes rather than selecting promising child nodes repeatedly. $\alpha\sqrt{\frac{2\ln N(V)}{N(v_i)}}$ is the exploration

term. If the visited times of node v_i is lower than other nodes, the exploration term becomes larger. Therefore, the tree can exploit promising nodes as well as exploring potential nodes

$$UCB1 = \frac{Q(v_i)}{N(v_i)} + \alpha\sqrt{\frac{2\ln N(V)}{N(v_i)}} \tag{3.11}$$

$Q(v_i)$: the reward value of node v_i
$N(v_i)$: the visited times of node v_i
$N(V)$: the visited times of root node V
α : a constant parameter in range 0 to 1

As long as the optimal branch or suboptimal branches haven't been discovered in early stages, the previous statistics are very misleading. It will spend too much time to explore those non-optimal branches. Consequently, besides the original UCT, we consider the sentiment of nodes. There are lots of unvisited child nodes in each round of selection because of the large search space and limited time. In order to find suboptimal paths in limited time, the rank of child nodes which have the same sentiment polarity to target node (previous selected node in upper layers) will be moved up, i.e., they have higher priority to be selected. The ranks of the rest of nodes are according to the polarity of target node. If the polarity of target node is positive or negative, the second group of polarity followed by the first group is neutral. We hope the sentiment polarity of child node is the same as target node as more as possible. Therefore, we put the neutral nodes in the middle of positive and negative nodes to separate.

We use sentiment dictionary instead of training a sentiment classifier. Combine

polarities by simple rules in table 3.4 if words are segmented. The polarity of segmented words are combined from the left to the right side iteratively until the last segmented one. The first column is the left side, and the first row is the right side. E.g., 保護 (P) 弱小 (N) → P, 假裝 (N) 乖巧 (P) → N, 出手 (N) 打人 (N) → N. The polarity is opposite if privatives are contained in segmented words, e.g., 從未 (V) 違規 (N) → P. There are still some words don't match rules, but the number of exceptions is acceptable.

We combine NTUSD [91], ANTUSD [92], CVAW2.0 [93] and 情感詞彙本體 [94] as our sentiment dictionary . Some of the words are in multiple classes, it may be an ambiguous word. We correct this type of problem so that one word only has one polarity. Besides these resources, we also expand sentiment dictionary by plesionyms. The size of sentiment dictionary is 22124 positive words, 26554 negative words and 27503 neutral words.

	P	N	T
P	P	P	P
N	N	N	N
T	P	N	T
V	N	P	T

Table 3.4: Combination rules of sentiments.

P : Positive
N : Negative
T : Neutral
V : Privative

Lastly, we select from top N child nodes which have higher scores. The N value is not fixed, it is inappropriate to select fixed number of nodes regardless of the children size, hence, it has different values according to the number of children.

Expansion

If the current node is a leaf node, the tree expands possible child nodes and chooses one of them randomly to simulate. Because not every child node is coherent to previous selected nodes in upper layers, we calculate the cosine similarity between child nodes and upper layer nodes. Child nodes won't be expanded if they are OOV or the cosine similarity score is less than zero. In order to reduce the computational cost, we expand current node only if the ratio of valid child nodes to invalid ones is less than 0.3. The current node will be removed if it can't expand child nodes (no connected concept in ConceptNet) or it is an invalid node. The nodes in upper layers will be removed too if they are not terminal nodes and don't have any child node. The dead end path in a tree is unnecessary. We don't expand child nodes which have the same concept as selected nodes in upper layers to prevent repeated or redundant concepts in a generated paragraph.

Simulation

MCTS runs a simulation from current node to terminal node by a default policy which is selecting nodes randomly according to paragraph templates. Simulate randomly is quick and simple. It doesn't need any domain knowledge and it can cover different regions of search tree.

However, it needs more time to converge a better result. It needs more time to simulate in early stages (step $<$ Nth step) of simulation because there are lots of unvisited nodes, hence the simulated time is relatively low for each node. The inaccurate predictions caused by low simulated times may bias the result of

selection if simulating randomly. The impact of errors in early stages is bigger than late stages because the selected nodes in late stages are based on upper layer nodes in the same path. Most of the nodes can be simulated in late stages of simulation because the number of unvisited nodes is low. The predictions are more accurate than early stages as the visited times increasing. In order to prevent the inaccurate simulation in early stages, we simulate with domain knowledge by word embedding. We adopt two methods here.

- The concepts which are more related (higher cosine similarity score) to target concept have higher probability to be simulated. For example, I go to fast food restaurant to eat fried chicken. The food is healthy. It's unlikely to say that fried chicken is a healthy food. The slot would be more reasonable by calculating the relatedness to other concepts.

- If the simulated concept is the third node in a sentence, we use the first and the second concept to predict the third concept by word embedding. The third concept which is related to the first two concepts has higher probability to be simulated. This is similar to sentence cloze test. The participants are asked to fill in the empty slot given other words in a sentence, e.g., I went to __ to buy fast food. The slot may be McDonald's or other fast food restaurants.

The advantages of these two methods are that they can avoid simulating unreasonable moves to reduce the computational cost. It needs less time to converge a better result, and have more time to explore or to exploit other nodes. Increase the accuracy of simulation by increasing the visited times in finite time. The dis-

advantages of them are that they need to train a word embedding and may not simulate some potential nodes since the word embedding is not 100 % accurate. The scores of some nodes are zero or even negative since they may not related to target concept or first two concepts in a sentence.

In order not to miss every possible node, we transform the distribution of scores. We don't use the cosine similarity scores or the probability of predicted concepts directly. First, we normalize the scores so that the summation is 1, and find a minimal value. Add the minimal value ($\log 0$ is undefined) and apply log transformation to make right-skewed distribution data (data concentrate on left side) more normal distribution. We then shift the data to positive by adding minimal value + 0.1 to avoid zero scores. Lastly, normalize the distribution again after log transformation. We don't want the highest related nodes to be simulated over and over, and ignore the low related nodes. As a result, we decrease the probability of maximal value and increase the probability of minimal value. It becomes more normal distribution than the original one.

The reward value of terminal node is not a simple binary representation which is winning (1) or losing (0) of a game. It's hard and complicated to evaluate how coherent a paragraph is. We use our coherence model instead of binary representation which is introduced in section 3.3. The coherent scores of terminal nodes will be evaluated at each time of simulation.

If concepts don't have any connected concepts, we use backup sentence templates to replace the current ones. If concepts still can't find any connected concepts, we change the paragraph template or initial concept by plesionyms. All of the information (parent, children, value, relation, reward value and visited times)

stored in each node will be reset, and restart the new MCTS round.

Backpropagation

Update $Q'(v_i)$ to simulated reward value $Q(v_i)$ of each node from current to root node in the traversed path. The visited times of nodes $N(v_i)$ in the path is incremented by 1.

$$s_i^{update} = \frac{Q(v_i) + Q'(v_i)}{N(v_i) + 1} \tag{3.12}$$

We reuse the subtree information in past time-steps of searching at subsequent time-steps. A node will be selected after a round of MCTS. This node is the new root node for the next round. We retain the subtree statistics below this new root and discard its parent node and the remainder of other sibling subtrees.

Chapter 4

Experiments and Results

4.1 ConceptNet Data Cleaning

In order to generate coherent paragraphs, the quality of input data is as important as the model. Although the coverage of original ConceptNet is wider than the modified one, it contain numerous errors. We refine the data, expand by different kinds of relations (mainly "Synonym") and ensure the quality of ConceptNet.

We add data of relation "Antonym" and "Synonym" from Chinese WordNet [1] [95], MOE revised dictionary [2] and manually. The number of each sources is shown in table 4.1 and 4.2. Use "Antonym" to expand "HasProperty", "Desires" and "NotDesires", and use "Synonym" to expand all relations.

	ConceptNet	Chinese WordNet	MOE revised dictionary
Size	31	100	9,146
Expanded size	HasProperty	Desires and NotDesires	
	2,650	830	

Table 4.1: Number of data from different sources and expanded size in Antonym.

[1] http://lope.linguistics.ntu.edu.tw/cwn/download/

[2] 中華民國教育部 (Ministry of Education, R.O.C.) « 重編國語辭典修訂本 » (版本編號：第五版) site:http://dict.revised.moe.edu.tw/

	ConceptNet	Chinese WordNet	MOE revised dictionary	Manually
size	1,018	1,700	9,756	2,511
Expanded size	844,448			

Table 4.2: Number of data from different sources and expanded size in Synonym.

Expanded ConceptNet is 3.21 times of the size of the original one. The actual number of each statistic in original ConceptNet should be smaller, because numerous incorrect concepts are included. The comparisons of data cleaning between original, modified and expanded ConceptNet are shown in table 4.3.

The words can be more precise if being represented by word embedding because the number of multiple segmented (> 1) concepts decreases up to 44%. We can see the significant difference of distribution in Figure 4.1. Though the number of distinct segmented concepts in the expanded ConceptNet is lower than the original one, we still have a much higher concept degree (Figure 4.2) which means more relations between concepts.

We unify low-degree concepts to other ones that is described in section 3.1.2 to increase the number of active concepts (degree > 3) and avoid spending time to explore dead end concepts.

CKB	CKB size	Distinct concept	Distinct segmented words
Original CN	352,411	121,606	1:30,692, 2:55,288, 3:10,720, 4:786
Modified CN	287,582	64,910	1:27,271, 2:33,801, 3:3,670, 4:60
Expanded CN	1,132,030	68,765	1:31,234, 2:33,801, 3:3,670, 4:60

	Active concepts	Average degree
Original CN	23,174(19.1%)	5.7
Modified CN	16,723(25.8%)	8.1
Expanded CN	33,131(48.1%)	32.2

CN: Chinese ConceptNet
Active concepts: concepts degree > 3

Table 4.3: Data cleaning comparison.

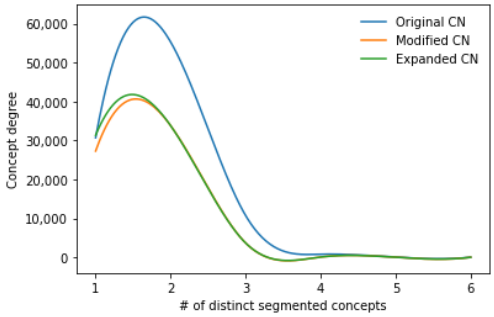


Figure 4.1: Distinct segmented concepts distribution.

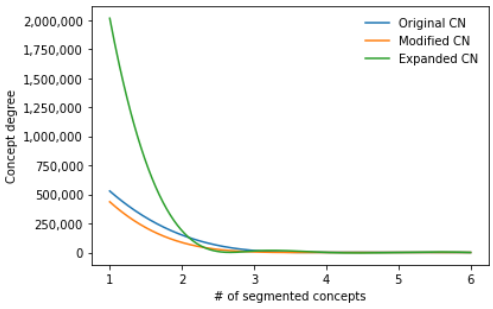


Figure 4.2: Segmented concepts distribution.

Table 4.4 shows the number of modifications, deletions and new data. The total number of modification and deletion is 325,417 which accounting for 92.3% in original ConceptNet.

Add			
Increased degrees	New concepts	Degree of new concepts	
88,927	29,519	66,191	
Modify			Delete
Concept pairs	Relations	SurfaceText	
142,939	31,463	51,959	99,056

Increased degrees: the number of increased degrees of concept if it exists in both original and modified ConceptNet.

Table 4.4: The number of data cleaning.

4.2 Word embedding

In this section, we experiment different hyperparameters of count-based and prediction-based word embeddings. We also compare our word embeddings to other pre-trained ones.

Evaluation

There are two methods to evaluate the quality of word embedding which are intrinsic and extrinsic evaluation. Intrinsic evaluation measures the semantic relatedness of the word embeddings by intermediate subtasks, such as analogy, categorization and relatedness. It's a simple and quick way to compute the score as the performance. And extrinsic evaluation uses word embedding as features to downstream NLP tasks, such as sentiment analysis, machine translation and Named Entity Recognition (NER) and observe the performance of tasks. Different tasks may favor different embeddings, the embeddings have a great result in the task may not have the same result in other tasks. It's a time-consuming evaluation, because we want to assess the relatedness between concepts over a large corpus, we choose similarity/relatedness tasks of intrinsic evaluation as our evaluation task.

We adopt Spearman's rank correlation coefficient in (4.1) as our evaluation function. It calculates the rank difference of two sequences in descending order. Two sequences are highly correlated if the value is close to 1, and less correlated if it is close to -1.

$$r_s = 1 - \frac{6\Sigma_{i=1}^{n} d_i^2}{n(n^2-1)}, -1 \leq r_s \leq 1 \qquad \begin{array}{l} d : \text{difference of two ranks} \\ n : \text{number of word pairs} \end{array} \tag{4.1}$$

We then compare scores with gold standards (human judgement) which are PKU-500 [96], SimLex-999 [97], WordSim-353 similarity (WS-353_sim) [98], Bruni MEN-3000 (MEN-3k) [99], Radinsky MTurk (MTurk-287) [100], WordSim-353 relatedness (WS-353_rel) [98], WordSim-240 (WS-240) [101] and WordSim-353 (WS-353) [102] is the combination of WS-353_sim and WS-353_rel. The Chinese

version of these datasets except PKU-500 and WS-240 are translated by [103]. We divide these datasets to similarity and relatedness datasets after examining the concept pairs.

	Similarity			Relatedness				Combined
Dataset	PKU	SimLex	WS353_sim	MEN	MTurk	WS353_rel	WS240	WS353
Size	494	999	203	2985	287	252	240	353

Table 4.5: Chinese gold standards of concepts similarity and relatedness.

Hyperparameters Comparison

The hyperparameters of count-based word embedding are shown in table 4.6a, and prediction-based hyperparameters are in table 4.6b. We experiment with varying hyperparameters and compare their performance in terms of Spearman correlation. Because the word embedding is used to calculate the relatedness between concepts in our research, the experimental results shown in figures are almost relatedness datasets. The hyperparameters used in each experiment are shown in the lower left of the figure.

As we described in the previous chapter, discovering new words by HMM-based model may find lots of meaningless words. We exclude words with frequency less than 80 in HMM-based model, and less than 30 in non-HMM based model. Despite the low frequency words are excluded, the HMM-based model still performs worse than the non-HMM based one as we expected in Figure 4.3. Therefore, we use non-HMM based model to segment words in both count-based and prediction-based model.

Count-based

Hyperparameter	Experimental values
Frequency weighting	Raw frequency, PMI variants
Window size	1~8
Dimensions	100~1200
Remove first k dimensions	k
Weighting exponent p	-1.5~1.5
Discover new words	yes, no

PMI variants: PMI, PPMI, NPMI, PMI^2, SPPMI

(a) Count-based.

Hyperparameter	Experimental values
Window size	1~8
Dimensions	100~1500
Model	Skip_gram, CBOW
Learning rate (LR)	0.025, 0.05
Sampling rate (SR)	0.1~0.00001
Negative samples (NS)	2, 5, 10
Discover new words	yes, no

(b) Prediction-based.

Table 4.6: Model hyperparameters.

Linear distance weighting The result of linear distance weighting in a fixed context window size is a little bit better than the raw frequency. The average Spearman scores in 8 datasets are 0.283 and 0.279 respectively in count-based model.

Frequency weighting The Figure 4.4 shows that co-occurrence matrix with raw frequency has poor performance than the other weighted methods. Among all the PMI variants, SPPMI performs the best. In the rest of the experiments, frequency weighting is using SPPMI.

Window size Small context window size is better than large one in similarity and relatedness datasets (Figure 4.5). It captures more associations between the target word and the context words, A large window size captures more topics

and domain information, as a result, it performs well in analogy tasks.

Dimensions The result of dimensions are in Figure 4.6. The performance becomes slowly increased when the dimensions is bigger than 500. In our count-based experiments, the performance stops increasing when dimensions $>$ 1000. This result is the same with [104]. They also found that the performance decreases when dimension from 500 to 1000 in count-based model.

Remove the first k dimensions In Figure 4.7, the performance of original SVD matrix is lower than removing the first 100 dimensions and the same as removing the first 150 dimensions (not in the figure). The best k is in range 5~10, starts decreasing after dimensions 10. In the similarity tasks, the best k is in the range 20~30 which is different from the relatedness tasks.

Weighting exponent The weighting exponent p in original SVD is 1. From Figure 4.8, the performance of p $>$ 1 is worse than the others. This result is found to be in line with the removing first k dimensions. They both reduce the impact of large singular values (which may contain noise) by removing or reducing them.

Although the best hyperparameters depends on different corpora, their importance can still be divided into three groups as, [Frequency weighting, remove first k dimensions, weighting exponent p], [dimensions, window size, HMM], [linear distance weighting] The best setting in the first group can increase the performance over 10 %. The second group is 3 % ~10 %, while the last group is less than 1 %.

Figure 4.3: Discover new words with or without HMM in count-based model.

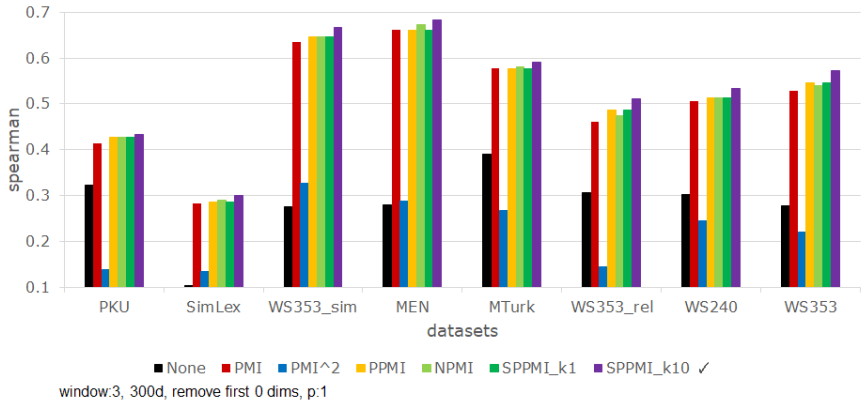


Figure 4.4: Different frequency weighting methods.

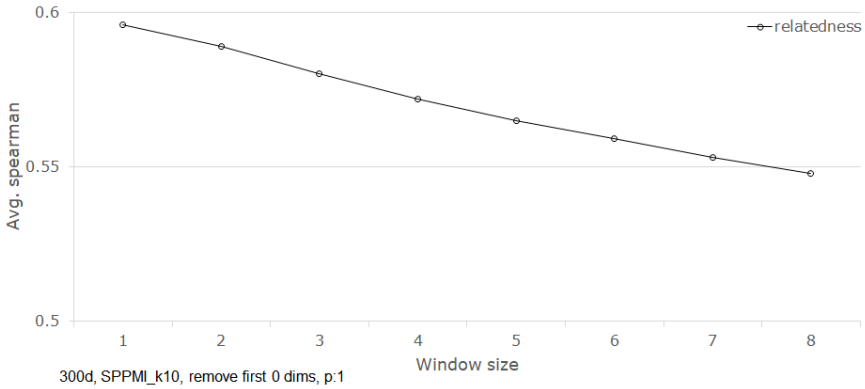


Figure 4.5: Window size.

The experimental result of prediction-based model are shown as figures in appendix C. The context window size and number of dimensions doesn't seem to affect the performance. Different negative samples have similar performance. However, the more negative samples in each training iteration, the more time it spends (appendix Figure 1h). In order to save training time, small window size,

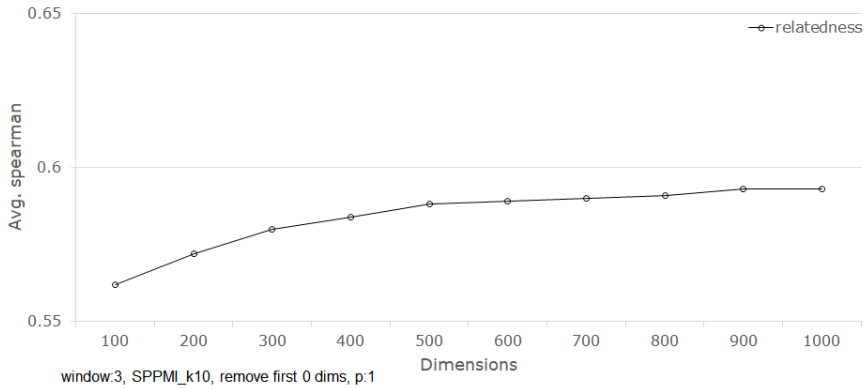


Figure 4.6: Dimensions.

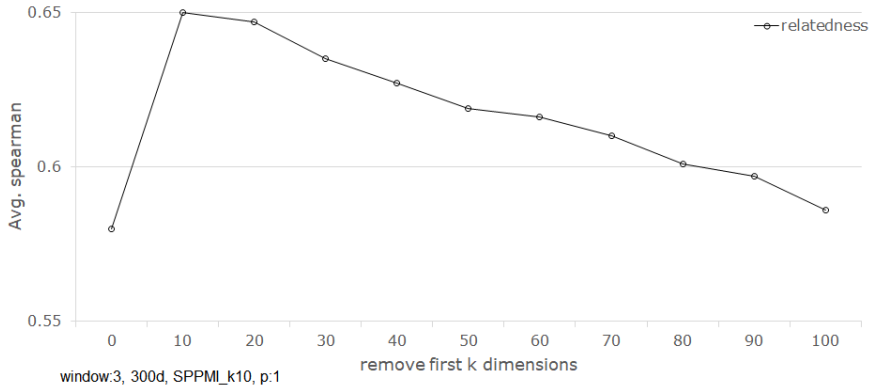


Figure 4.7: Remove first k dimensions.

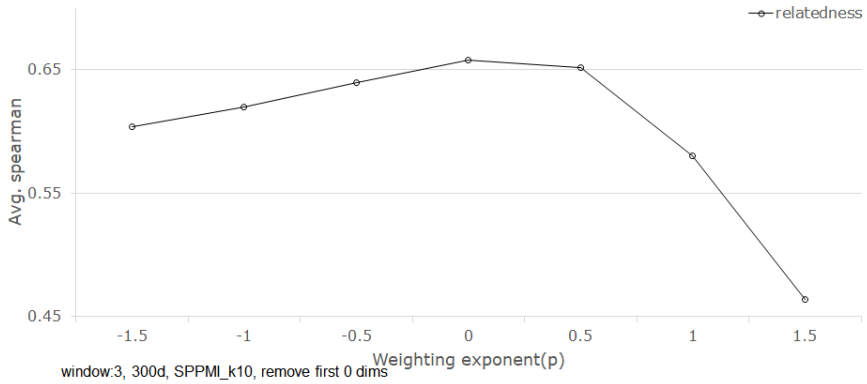


Figure 4.8: Weighting exponent.

dimensions and negative samples are recommended. Skip-gram model performs better than CBOW in the relatedness tasks, but worse in the similarity tasks.

The best settings of hyperparameters and performance are in Figure 4.9 and table 4.7, the best settings in similarity task are in appendix table A.6.

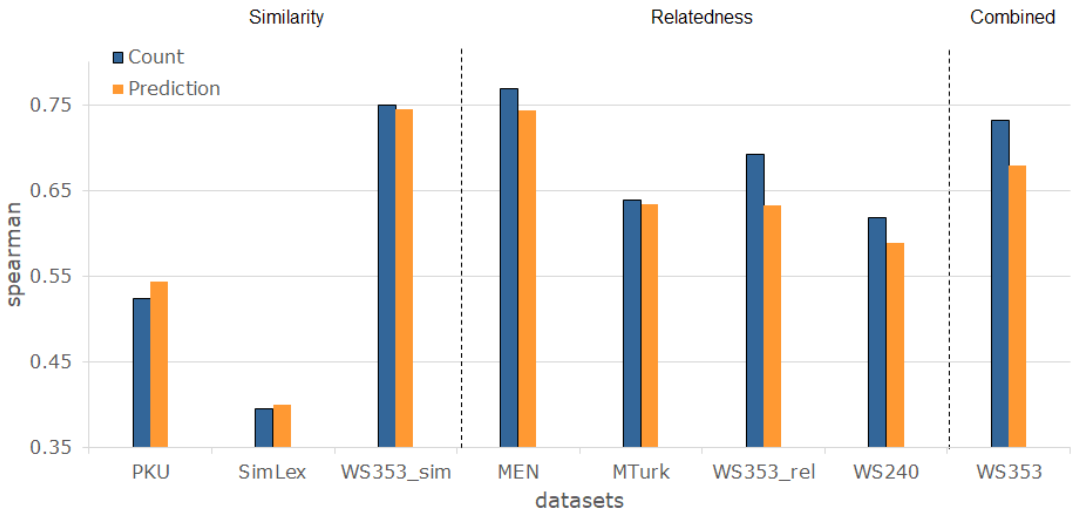


Figure 4.9: Best settings of hyperparameters in count-based and prediction-based model.

Hyperparameter	best settings
Frequency weighting	SPPMI_k10
Window size	3
Dimensions	700
Remove first k dimensions	6
Weighting exponent p	0.5
Discover new words	no

(a) Count-based.

Hyperparameter	best settings
Window size	2
Dimensions	500
Model	skip_gram
Learning rate (LR)	0.05
Sampling rate (SR)	0.00001
Negative samples (NS)	2
Discover new words	no

(b) Prediction-based.

Table 4.7: Best hyperparameter settings in relatedness task.

Comparison with Other Pre-trained Word Embedding Methods

The information of our and other pre-trained word embeddings are shown in table 4.8. The vocabulary size of fastText_cc is much lower than 2,000,000. It contains English, symbols, numbers and simplified Chinese. All of the pre-trained word embeddings are traditional Chinese except fastText_cc and Numberbatch. The vocabularies in Numberbatch are different from other models. They are concepts not words because they train on ConceptNet which consists of concepts.

Word embedding	Dim	Vocabulary size	Dataset	Model
Our WE model_count	700	249,571	PTT,Wiki	count-based
Our WE model_prediction	500	249,571	PTT,Wiki	skip-gram, CBOW
CKIP_Glove[3]	300	480,551	TCNAGC,ASBC	Glove
CKIP_word2vec	300	473,202	TCNAGC,ASBC	word2vec
fastText_wiki[4] [105]	300	141,676	Wiki	CBOW
fastText_cc[5] [106]	300	2,000,000	CC, Wiki	skip_gram
Numberbatch[6] [5]	300	307 441	CN	CN,word2vec,Glove

Dim : Dimension
WE : Word embedding
CKIP : Chinese Knowledge and Information Processing
TCNAGC : Taiwan's Central News Agency Gigaword Corpus
ASBC : Academia Sinica Balanced Corpus
CC : Common Crawl
CN : ConceptNet

Table 4.8: Pre-trained word embeddings information.

Table 4.9 shows comparison of our word embeddings and other pre-trained ones in relatedness tasks (the result of similarity task is on appendix table A.4). Our count-based model outperforms other models in the similarity/relatedness tasks, and our prediction-based model and Numberbatch are the second. Even if the same dimensions (300d), our model still outperforms other pre-trained ones.

From the table 4.10, the average OOV (vocabulary coverage) of our model is almost the same as fastText_cc despite ours contains less vocabularies With proper corpora pre-processing and some weighting techniques, count-based model can still outperform prediction-based models (except Numberbatch). Besides these pre-trained models, I would

to mention experiments implemented by [107]. They compared the dynamic mod-

[3] https://ckip.iis.sinica.edu.tw/project/embedding
[4] https://fasttext.cc/docs/en/crawl-vectors.html
[5] https://fasttext.cc/docs/en/pretrained-vectors.html
[6] https://github.com/commonsense/conceptnet-numberbatch

els (Elmo, GPT2, BERT) with the static models (skip-gram, CBOW, Glove, fastText) on the similarity tasks. The datasets they used are MEN, WS353_sim, WS353_rel, WS353 and SimLex which are similar to ours (their datasets language is English). They found that dynamic models are not superior to static models.

Word embedding	Dim	Relatedness				Avg.	Combined
		MEN	MTurk	WS353_rel	WS240		WS353
Our WE model_count	700	.769	.639	.692	.618	**.679**	.732
Our WE model_count	300	.757	.631	.663	.609	.665	.709
Our WE model_prediction	500	.744	.634	.633	.589	.650	.669
Our WE model_prediction	300	.744	.634	.632	.583	.648	.685
CKIP_Glove	300	.622	.549	.472	.556	.550	.516
CKIP_word2vec	300	.698	.614	.612	.578	.626	.657
fastText_wiki	300	.371	.351	.091	.277	.272	.242
fastText_cc	300	.624	.574	.546	.480	.556	.608
Numberbatch	300	.758	.703	.636	.538	.659	.700

Table 4.9: Compare to other pre-trained word embeddings on relatedness datasets.

Word embedding	Vocab size	Relatedness				Combined	Avg.(%)
		MEN	MTurk	WS353_rel	WS240	WS353	
Our model	249,571	76	46	15	7	23	**4.1**
CKIP	480,551	129	43	10	16	19	5.3
fastText_wiki	50,184	328	78	27	20	48	12.2
fastText_cc	307,441	77	37	9	7	16	**3.6**
Numberbatch	141,676	483	110	46	24	65	17.7

Table 4.10: Compare to other pre-trained word embeddings (OOV).

4.3 Discourse Coherence Model

We compare different hyperparameters of DNN coherence model in this section. Each paragraph consists of 4 or 5 sentences, and the max number of words in a sentence is 10. The number of articles in training, validation and test data is

559537, 112279, 107002 (7:1.5:1.5). (The less important experiments are shown in appendix.)

We first evaluate whether the word embedding affects the training or not. Table 4.11 shows the comparison of joint trained and non-joint trained word embeddings. The accuracy of joint trained word embedding is lower than non-joint trained model, and the training is slower too. And in table 4.12, the word embedding with higher Spearman score has higher accuracy than the other one.

WE joint trained	Train		Validation		Test		time/epoch
	loss	acc	loss	acc	loss	acc	
Yes	.277	.821	.410	.803	.448	.785	162 mins
No	.088	.972	.214	.929	.305	**.902**	**138** mins

Table 4.11: Word embedding joint trained.

Spearman score of WE	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
.604	.376	.749	.366	.763	.370	.760
.681	.263	.835	.348	.783	.327	**.800**

Table 4.12: Different scores of word embeddings.

The number of words in a sentence and how bidirectional model merges forward and backward hidden states doesn't seem to affect the performance. The results are shown in appendix table A.9

Table 4.13 shows the experiment of whether paragraphs with duplicate sentences will affect the performance or not. The size of moving sentence window is 1. The next sample is the original samples moving down by one sentence. The accuracy in both are pretty much the same in training samples, but the duplicate one perform worse in validation and test samples, which means training samples

without duplicate sentences can handle more different kinds of data. Replaced

Duplicate sentence	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
Yes	.216	.916	.291	.885	.366	.860
No	.189	.928	.151	.944	.142	**.947**

Table 4.13: Paragraphs with duplicate sentences or not.

rate is the probability of replacing context in negative samples. Each sentence in a paragraph has x% to be replaced. In table 4.14, the more concepts replaced, the higher the accuracy. It's also intuitive to know that a paragraph is less coherent if it dissimilars more from the original one. From table 4.15 and 4.17, we can know that low replaced rate model can handle high replaced rate test data, but not vice versa.

Replaced rate	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
.35	.260	.828	.297	.813	.306	.806
.50	.251	.828	.279	.816	.288	.808
.65	.235	.836	.250	.826	.256	.822
.80	.247	.825	.245	.827	.239	.831

Table 4.14: Compare different replaced rate of negative samples.

Replaced rate	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
.25	.204	.925	.147	.946	.136	.950

Table 4.15: Training replaced rate 0.25.

Replaced rate	Test	
	loss	acc
.10	.536	.790
.20	.197	.925
.30	.112	.962
.40	.092	.971
.50	.086	.972
.80	.086	.973

Table 4.16: Replaced rate 0.25 in different test data.

Replaced rate	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
.65	.041	.989	.012	.997	.012	.997

Table 4.17: Training replaced rate 0.65.

Replaced rate	Test	
	loss	acc
.10	2.19	.541
.20	1.29	.679
.30	.558	.840
.40	.173	.945
.50	.043	.986

Table 4.18: Replaced rate 0.65 in different test data.

In table 4.19, we can see the accuracy of replacing with arbitrary concepts is higher than replacing with connected concepts in ConceptNet. It means the model can distinguish the coherent or incoherent paragraphs easily if negative samples are replaced by arbitrary concepts. Intuitively we know that paragraphs being replaced by arbitrary concepts are obvious incoherent. Our MCTS-based model selects concepts from the connected ones. If the negative samples are replaced by arbitrary concepts, the performance isn't very well. All the scores of paragraphs generated by MCTS-based model are over 0.95. Namely, it can't distinguish between the good or bad ones. Consequently, we make negative samples being replaced with connected concepts. The negative samples in preceding experiments are replaced by arbitrary concepts, and in the following experiments they are replaced with connected concepts.

Negative samples	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
Arbitrary concepts	.329	.859	.268	.889	.271	.888
Connected concepts	.501	.757	.473	.779	.470	.777

Table 4.19: Negative samples replaced by arbitrary concepts against the connected ones.

In table 4.20, BiLSTM and BiGRU are almost the same, and both of them are

a bit better than LSTM and GRU. The small batch size performs better than the large one, but it needs more time to train. We select 64 as our batch size. The model performs better when we use more hidden units in general.

NN architecture	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
BiRNN	.526	.737	.531	.735	.528	.739
BiLSTM	.286	.880	.308	.873	.309	**.873**
BiGRU	.285	.881	.316	.871	.315	**.871**
RNN	.541	.726	.526	.737	.529	.738
LSTM	.324	.861	.339	.857	.341	.857
GRU	.326	.860	.335	.857	.333	.859

Table 4.20: NN architectures.

Batch size	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
16	.491	.763	.446	.791	.447	**.790**
32	.501	.757	.473	.779	.470	.777
64	.508	.756	.454	.790	.456	**.788**
128	.519	.744	.477	.776	.465	.779
256	.534	.735	.491	.764	.482	.772

Table 4.21: Batch size.

Hidden units	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
8	.540	.730	.517	.750	.509	.753
16	.503	.756	.463	.781	.463	.782
32	.463	.782	.422	.807	.415	.810
64	.433	.800	.391	.824	.392	.825
128	.410	.813	.379	.836	.379	.834
256	.398	.820	.360	.842	.361	**.842**

Table 4.22: Hidden units.

Optimizer	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
SGD	.557	.712	.534	.736	.540	.733
RMSprop	.423	.811	.389	.834	.382	.835
Adam	.385	.828	.353	.846	.350	**.850**
Adagrad	.637	.639	.601	.683	.600	.683

Table 4.23: Optimizers.

Learning rate	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
.1	.696	.500	.693	.501	.695	.500
.01	.638	.641	.625	.663	.658	.614
.001	.503	.756	.463	.781	.463	**.782**
.0001	.569	.708	.520	.746	.521	.745
.00001	.674	.583	.654	.611	.654	.612

Table 4.24: learning rate.

Architecture	Epoch	Batch size	Batch numbers	Hidden units	Optimizer	Learning rate	Dropout rate
BiLSTM	10	64	16801	256	Adam	.001	.2

Table 4.25: Best settings of hyperparameters.

4.3.1 Examples

We show some examples of low and high coherence score in table 4.26. The paragraphs on the left side have higher coherence than the right side. Four or five words in the first two examples are substituted by other connected words in ConceptNet. The coherence model can distinguish them easily. The rest of the examples only substitute two words, the model can still predict correctly.

In table 4.27, we replace the word manually. The paragraphs in the left side are replaced by plesionyms, and the ones in the right side are replaced by less related words. We find that the coherence score is the same or better than the original paragraph if replaced by plesionyms, and the score is lower than the original one if replaced by less related words. Although some of the scores are 0.7 and 0.8, they still lower than the original one. It proves that the coherence model can distinguish the coherent and incoherent paragraphs.

	High coherence	Low coherence
1	來到 角板山 行館 高處 **眺望** 溪口 台地 好 風光 美好 **景色** 一覽無遺 前往 **角板山** 樟腦 文化 特展 路上 經過 寧靜 **池子** 適合 爸媽 **走走** score:0.978	來到 角板山 行館 高處 **推下來** 溪口 台地 好 風光 美好 **人生** 一覽無遺 前往 **蟑螂** 樟腦 文化 特展 路上 經過 寧靜 **大海** 適合 爸媽 **拌嘴** score:0.004
2	父親節 家裡 **關係** 變 不好 父母 對 哥哥 **特別** 好 今天 **難得** 姐姐 休假 姐姐 提前 爸爸 約 **今天** 吃 父親節 大餐 哥哥 無法 排休 難 調班 score:0.994	父親節 家裡 **玩遊戲** 變 不好 父母 對 哥哥 **沮喪** 好 今天 **開心** 姐姐 休假 姐姐 提前 爸爸 約 **夜半** 吃 父親節 大餐 哥哥 無法 排休 難 調班 score:0.046
3	鄉民 推薦 三民區 三立 飯丸 該店 開業 招牌 **荷包蛋** 飯丸 口 咬下 流出 濃郁 蛋液 傳統 肉燥 菜脯 肉鬆 油條 佐料 加 豆皮 **香氣** 十足 包 滷蛋 大小 飯丸 score:0.787	鄉民 推薦 三民區 三立 飯丸 該店 開業 招牌 **店鋪** 飯丸 口 咬下 流出 濃郁 蛋液 傳統 肉燥 菜脯 肉鬆 油條 佐料 加 豆皮 **黃豆** 十足 包 滷蛋 大小 飯丸 score:0.543
4	物聯網 裝置 可貴 透過 終端 節點 蒐集到 資訊 經由 網路 回 雲端 進行 大數據 分析 **數據** 分析 有用 商業 政策 **發展** 寶貴 資訊 score:0.997	物聯網 裝置 可貴 透過 終端 節點 蒐集到 資訊 經由 網路 回 雲端 進行 大數據 分析 **名嘴** 分析 有用 商業 政策 **愛護** 寶貴 資訊 score:0.424
5	財團法人 高雄市 文武 聖殿 董事長 說 首度 試辦 愛心 餐券 清寒 小朋友 **平時** 在校 營養 午餐 寒假 期間 家長 全天 上班 無人 在家 怕 學生 **挨餓** score:0.721	財團法人 高雄市 文武 聖殿 董事長 說 首度 試辦 愛心 餐券 清寒 小朋友 **零分** 在校 營養 午餐 寒假 期間 家長 全天 上班 無人 在家 怕 學生 **上床睡覺** score:0.004

Table 4.26: Coherence model results on test dataset.

Original paragraph	
請 大家 幫忙 朋友 照顧 貓咪 凌晨 開 紗窗 出去 早上 起床 發現 不見了 請 大家 幫忙 注意 score:0.904	
Replaced by plesionyms	Replaced by less related word
請 大家 **協助** 朋友 照顧 貓咪 凌晨 開 紗窗 出去 早上 起床 發現 不見了 請 大家 幫忙 注意 score:0.969	請 大家 幫忙 朋友 照顧 貓咪 凌晨 開 紗窗 出去 早上 **天晴** 發現 不見了 請 大家 幫忙 注意 score:0.094
請 大家 幫忙 朋友 照顧 貓咪 凌晨 開 紗窗 出去 早上 起床 發現 不見了 請 大家 幫忙 **留意** score:0.906	請 大家 幫忙 朋友 照顧 貓咪 凌晨 開 紗窗 出去 早上 起床 發現 不見了 請 大家 幫忙 **水災** score:0.03
請 大家 **協助** 朋友 照顧 貓咪 凌晨 開 **窗戶** 出去 早上 **起身** 發現 不見了 請 大家 幫忙 **留意** score:0.955	請 大家 幫忙 朋友 照顧 貓咪 凌晨 開 紗窗 出去 早上 起床 發現 不見了 請 大家 幫忙 **超人** score:0.736
	請 大家 **吃飯** 朋友 照顧 貓咪 凌晨 開 紗窗 出去 早上 起床 發現 不見了 請 大家 幫忙 注意 score:0.806

Table 4.27: Coherence model results testing.

4.4 Paragraph Generation

We evaluate the generated paragraphs by human rating. We employee 30 web users to rate the paragraphs from 1 to 6 according to coherence, fluency and correctness. The paragraph with lower score is less coherent, and the one with higher score is more coherent.

We compare 4 different types of paragraphs, and each one has 7 different paragraphs based on different templates. The paragraphs are generated by human, our generation model with all sub-models and enhancements, our model without coherence evaluation and our model extracting commonsense knowledge from the original Chinese ConceptNet (without data cleaning). The results of human evaluation are shown in table 4.28.

	Human-written	Our model	Without coherence model	Original CN
Human rating	4.70	3.23	2.14	1.55
Human-written as gold standards	5	3.43	2.39	1.71

Table 4.28: The result of human rating.

We can see that the paragraphs generated by human get the highest score which is what we expected. The second one is paragraphs generated by our model. The third one is paragraphs generated by our model without coherence evaluation. The last one is extracting concepts from original ConceptNet. Most of the participants rate the last one as lowest score (1). There are two reasons why. First, original ConceptNet contains lots of errors and redundancy which we discussed in section 3.1. Secondly, the ConceptNet that we use to extract connective con-

cepts to replace the context words when training the coherence model is refined one, not the original one. Consequently, the coherence model doesn't work in the original ConceptNet. Also, we found that extracting concepts from the original ConceptNet are the most time-consuming method to generate paragraphs. The generated model can't find connective concepts, it has to change the initial concept or template over and over, and it still can't generate paragraphs successfully most of the time.

Because the human evaluation is subjective, everyone may have different opinions. Some participants even rate the human-written paragraphs as incoherent. Some people tend to rate higher scores, and some tend to rate lower scores. Therefore, the absolute scores are not precise. If we see the human-written paragraphs as gold standards, then we normalize the score from 1 to 5 (the second row of table 4.28). The score of our model is 3.43 which is over half of five. It means that the paragraphs generated by our model are coherent to some degree. The concepts in the paragraphs are combined closely way more than extracting concepts randomly in ConceptNet.

In table 4.29, we list some examples which are in the human evaluation dataset. The examples on left side are the highest score with that method, and the ones on the right side are the lowest score. The lowest score of our model (the second row in table 4.29) is 2.41. Participants may see the similar meaning of [優秀員工] and [好員工], hence they rate it as incoherent. We can see that the participants rate paragraphs to lower score even there are only one or two words that incoherent to other words.

A	學生上課時首先要拿筆作筆記 辛苦的讀書是為了將來 讀書時不能懶惰會令人落榜 可以邊讀邊聽音樂來沉澱心情 score: 5.11	朋友工作時懼怕出錯，會令人自責 他希望有錢時能買房子來過上快樂的生活 但沉重的房貸會帶來壓力 壓得他喘不過氣 score:4.59
B	黑道會為了耍帥而飆車 刺激的開快車時會躲避警察 耍酷時會抽菸來假裝成熟 危害健康的汙染會帶來損害 score:4.37	店裡的雇主是體恤員工的 老闆會照顧員工和賺大錢 想要優秀員工和好員工 是會令人勞累的職業 score:2.41
C	教師騎車時害怕摔車 摔車會帶來嘲諷 因為騎車而累時會喝保力達 是為了用心的精力 難過的睡不著時會想要打架 score:2.78	警方打架時要優先臭罵來動手 麻煩的拌嘴是為了確定答案 爭執時分手會令人生氣 離婚時移居是為了購屋 score:1.74
D	暖和的教師是有些很機車的 是會學習的職業 會令人想要就學、去上課 上學時會健康的吃早餐來出門買早餐 score:1.70	錢的老闆是討厭的的 老闆會給錢和性騷擾員工 想要開除員工和懶惰 是會難吃蛋餅的會罵你的人 score:1.19

A: human-written
B: our generation model
C: our generation model without coherence model as reward function in MCTS
D: extract concepts from original ConceptNet

Table 4.29: Some examples of generated paragraphs in human evaluation dataset.

Chapter 5

Conclusion

5.1 Summary

To generate coherent texts is very difficult for machines without correct commonsense knowledge. We found that inaccurate commonsense knowledge will mislead following applications such as reasoning or text generation. We adopt several approaches to refine the ConceptNet CKB, because it originated from the crowdsourcing which may have numerous errors. We extract commonsense knowledge from ConceptNet automatically to generate coherent paragraphs based on predesigned templates. The concepts are connected by some relations which provide semantic information in paragraphs. Unlike the traditional text generation is constrained by the problem of large search space in selecting associate concepts, we adopt MCTS to deal with this problem. Some techniques are adopted to reduce the computational cost and speed up the searching process, including word embedding and sentiment. The word embedding that we constructed outperform other pre-trained ones on relatedness task. We also expand ConceptNet by plesionyms so that the generated paragraphs are more versatile.

Different from traditional text generation evaluated their text output only by human, we evaluate our paragraphs objectively and subjectively. We evaluate generated paragraphs automatically while generating by a Deep Neural Network coherence model which is trained on six hundred thousand of articles collected from online forum and news, and web users evaluate paragraphs generated by different methods. From both subjective and objective evaluations, our model perform better than extracting concepts from original ConceptNet and random reward values when using the expanded and modified ConceptNet and discourse coherence model as our reward function of MCTS.

5.2 Future Work

The evaluations on ConceptNet data cleaning are not complete. We only show the the performance when generating paragraphs. There is another possible task to evaluate the quality of ConceptNet. Calculating the word embedding based on our ConceptNet and compare to original one on similarity/relatedness tasks. The word embedding trained on our ConceptNet should perform better on similarity/relatedness tasks since unreasonable data is removed. The efficiency of MCTS performance could be evaluated more rigorously. The expanded ConceptNet has a larger search space than the original one, it may spend much more time to converge. It doesn't guarantee our models can always get higher coherence in the generated text given relatively short time. However, ours may perform better than the original one given enough searching time in general.

To design of human-crafted sentences and paragraph templates is time-consuming

and lack adaptability and maintainability especially for paragraph templates. Therefore, paragraph templates can be replaced by random combinations of relations on the premise that the coherence model can evaluate coherence more accurately. Random combinations of relations could yield more unordered and unstructured text than the usage of tentatively designed paragraph templates. It needs a better coherence model to evaluate the performance. And the relations that won't be used in daily life should be filtered out; otherwise there will be lots of inappropriate combinations of relations. The usable relations in Chinese ConceptNet are still not enough (only twelve relations can be used). The paragraphs are constrained by the limited relations that lack of variation and flexibility. It can be extended by collecting more types of relations by GWAP crowd-sourcing and then refine them again. Once we don't need templates and have enough different relations, we can generate more concepts in a sentence and more number of sentence in a paragraph.

We didn't attempt other different MCTS strategies, such as ϵ-greedy, First Play Urgency (FPU) [108], UCB1-Tuned [109], etc. And more domain knowledge can be added in the simulation phase. Different strategies and domain knowledge may both increase the accuracy and reduce search time.

Bibliography

[1] H. Liu and P. Singh, "Makebelieve: using commonsense knowledge to generate stories," in *AAAI/IAAI*, (USA), pp. 957–958, American Association for Artificial Intelligence, 2002.

[2] S. Yu and E. Ong, "Using common-sense knowledge in generating stories," in *PRICAI 2012: Trends in Artificial Intelligence*, (Berlin, Heidelberg), pp. 838–843, Springer Berlin Heidelberg, 2012.

[3] P. Yang, F. Luo, P. Chen, L. Li, Z. Yin, X. He, and X. Sun, "Knowledgeable storyteller: A commonsense-driven generative model for visual storytelling," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 5356–5362, International Joint Conferences on Artificial Intelligence Organization, 2019.

[4] H. Zhang, Z. Liu, C. Xiong, and Z. Liu, "Grounded conversation generation as guided traverses in commonsense knowledge graphs," in *ACL 2020*, Association for Computational Linguistics, 2020.

[5] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 4444–4451, AAAI Press, 2017.

[6] R. Coulom, "Efficient selectivity and backup operators in monte-carlo tree search," in *Proceedings of the 5th International Conference on Computers and Games*, CG'06, (Berlin, Heidelberg), p. 72–83, Springer-Verlag, 2006.

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[8] B. Kartal, J. Koenig, and S. J. Guy, "User-driven narrative variation in large story domains using monte carlo tree search," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, (Richland, SC), p. 69 – 76, International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[9] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 2157–2169, Association for Computational Linguistics, 2017.

[10] K. Kumagai, I. Kobayashi, D. Mochihashi, H. Asoh, T. Nakamura, , and T. Nagai, "Natural language generation using monte carlo tree search," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 22, no. 5, pp. 777–785, 2018.

[11] S. Mukherjee, "An unsupervised approach to automatic response generation for conversational e-commerce agents using monte carlo tree search," 2019.

[12] D. B. Lenat and R. V. Guha, *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.

[13] C. Fellbaum, "Wordnet: an electronic lexical database," *Language*, vol. 76, p. 706, 2000.

[14] N. Tandon, G. de Melo, F. Suchanek, and G. Weikum, "Webchild: harvesting and organizing commonsense knowledge from the web," in *WSDM*, (New York, NY, USA), pp. 523–532, Association for Computing Machinery, 2014.

[15] J. Romero, S. Razniewski, K. Pal, J. Z. Pan, A. Sakhadeo, and G. Weikum, "Commonsense properties from query logs and question answering forums," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, (New York, NY, USA), p. 1411–1420, Association for Computing Machinery, 2019.

[16] M. Sap, R. LeBras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi, "Atomic: An atlas of machine commonsense for if-then reasoning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3027–3035, 2019.

[17] J. Gordon and B. Durme, "Reporting bias and knowledge acquisition," in *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, (New York, NY, USA), pp. 25–30, Association for Computing Machinery, 2013.

[18] B. Van Durme, *Extracting implicit knowledge from text*. PhD thesis, University of Rochester, Rochester, NY 14627, 2010.

[19] J. F. Sowa, *Semantic Networks*. Encyclopedia of Artificial Intelligence, Shapiro, S.C. (Ed.). New York: Wiley and Sons, 1987.

[20] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. Li Zhu, "Open mind common sense: Knowledge acquisition from the general public," in *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, vol. 2519, (Berlin, Heidelberg), pp. 1223–1237, Springer Berlin Heidelberg, 2002.

[21] Y.-L. Kuo, J.-C. Lee, K.-Y. Chiang, R. Wang, E. Shen, C. wei Chan, and J. Y.-J. Hsu, "Community-based game design: Experiments on social games for commonsense data collection," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, (New York, NY, USA), p. 15–22, Association for Computing Machinery, 2009.

[22] L. v. Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, p. 92 – 94, 2006.

[23] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 17–30, 1989.

[24] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, ACL '94, (USA), p. 133 – 138, Association for Computational Linguistics, 1994.

[25] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, (San Francisco, CA, USA), p. 448–453, Morgan Kaufmann Publishers Inc., 1995.

[26] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, (San Francisco, CA, USA), p. 296–304, Morgan Kaufmann Publishers Inc., 1998.

[27] G. Hirst and D. St-Onge, *Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms*, pp. 305–332. The MIT Press, 1998.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, Curran Associates, Inc., 2013.

[29] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, 2018.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long*

and Short Papers), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, 2019.

[32] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Baltimore, Maryland), pp. 302–308, Association for Computational Linguistics, 2014.

[33] W. Che, Y. Shao, T. Liu, and Y. Ding, "SemEval-2016 task 9: Chinese semantic dependency parsing," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, (San Diego, California), pp. 1074–1080, Association for Computational Linguistics, 2016.

[34] M. A. K. Halliday and R. Hasan, *Cohesion in English*. London:Longman, 1976.

[35] R. Kibble and R. Power, "Optimizing referential coherence in text generation," *Comput. Linguist.*, vol. 30, no. 4, p. 401–416, 2004.

[36] R. Soricut and D. Marcu, "Discourse generation using utility-trained coherence models," in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, (Sydney, Australia), pp. 803–810, Association for Computational Linguistics, 2006.

[37] M. Lapata, "Probabilistic text structuring: Experiments with sentence ordering," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, ACL '03, (USA), pp. 545–552, Association for Computational Linguistics, 2003.

[38] N. Okazaki, Y. Matsuo, and M. Ishizuka, "Coherent arrangement of sentences extracted from multiple newspaper articles," *Lecture Notes in Computer Science*, vol. 3157, pp. 882–891, 2004.

[39] D. Bollegala, N. Okazaki, and M. Ishizuka, "A bottom-up approach to sentence ordering for multi-document summarization," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, (Sydney, Australia), pp. 385–392, Association for Computational Linguistics, 2006.

[40] E. Miltsakaki and K. Kukich, "Automated evaluation of coherence in student essays," in *In Proceedings of LREC 2000*, pp. 1–8, 2000.

[41] D. Higgins, J. Burstein, D. Marcu, and C. Gentile, "Evaluating multiple aspects of coherence in student essays," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, (Boston, Massachusetts, USA), pp. 185–192, Association for Computational Linguistics, 2004.

[42] J. Hobbs, "On the coherence and structure of discourse," tech. rep., Center for the Study of language and Information, Stanford University, 1985.

[43] J. C. Lester and B. W. Porter, "Developing and empirically evaluating robust explanation generators: The knight experiments," *Comput. Linguist.*, vol. 23, no. 1, p. 65–101, 1997.

[44] R. M. Young, "Using grice's maxim of quantity to select the content of plan descriptions," *Artificial Intelligence*, vol. 115, pp. 215–256, 1999.

[45] A. S. Law, Y. Freer, J. Hunter, R. Logie, N. McIntosh, and J. Quinn, "A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit," *Journal of Clinical Monitoring and Computing*, vol. 19, pp. 183–194, 2005.

[46] P. Engelhardt, K. Bailey, and F. Ferreira, "Do speakers and listeners observe the gricean maxim of quantity?," *Journal of Memory and Language*, vol. 54, no. 4, pp. 554–573, 2006.

[47] M. Elsner, J. Austerweil, and E. Charniak, "A unified local and global model for discourse coherence," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, (Rochester, New York), pp. 436–443, Association for Computational Linguistics, 2007.

[48] R. Barzilay and M. Lapata, "Modeling local coherence: An entity-based approach," *Computational Linguistics*, vol. 34, no. 1, pp. 1–34, 2008.

[49] S. Bangalore, O. Rambow, and S. Whittaker, "Evaluation metrics for generation," in *Proceedings of the First International Conference on Natural Language Generation - Volume 14*, INLG '00, (USA), p. 1–8, Association for Computational Linguistics, 2000.

[50] I. Langkilde-Geary, "An empirical verification of coverage and correctness for a general-purpose sentence generator," in *Proceedings of the 2nd International Conference on Natural Language Generation*, (Harriman, New York, USA), pp. 17–24, Association for Computational Linguistics, 2002.

[51] A. Belz and E. Reiter, "Comparing automatic and human evaluation of NLG systems," in *11th Conference of the European Chapter of the Association for Computational Linguistics*, (Trento, Italy), pp. 313–320, Association for Computational Linguistics, 2006.

[52] J. Li and E. Hovy, "A model of coherence based on distributed sentence representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 2039–2048, Association for Computational Linguistics, 2014.

[53] J. Li and D. Jurafsky, "Neural net models of open-domain discourse coherence," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 198–209, Association for Computational Linguistics, 2017.

[54] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32 of *Proceedings of Machine Learning Research*, (Bejing, China), pp. 1188–1196, PMLR, 2014.

[55] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, (Cambridge, MA, USA), p. 3294–3302, MIT Press, 2015.

[56] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *ICLR*, 2017.

[57] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," 2017.

[58] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal sentence encoder for English," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Brussels, Belgium), pp. 169–174, Association for Computational Linguistics, 2018.

[59] O. Dušek, D. M. Howcroft, and V. Rieser, "Semantic noise matters for neural natural language generation," in *Proceedings of the 12th International Conference on Natural Language Generation*, (Tokyo, Japan), pp. 421–426, Association for Computational Linguistics, 2019.

[60] V. W. Soo, T.-Y. Lai, K.-J. Wu, and Y.-P. Hsu, "Generate modern style chinese poems based on common sense and evolutionary computation," *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 315–322, 2015.

[61] X. Feng, M. Liu, J. Liu, B. Qin, Y. Sun, and T. Liu, "Topic-to-essay generation with neural networks," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 4078–4084, International Joint Conferences on Artificial Intelligence Organization, 2018.

[62] M. Elhadad and J. Robin, "An overview of SURGE: a reusable comprehensive syntactic realization component," in *Eighth International Natural Language Generation Workshop (Posters and Demonstrations)*, 1996.

[63] J. Coch, "Overview of AlethGen," in *8th International Natural Language Generation Workshop (Posters and Demonstrations)*, 1996.

[64] B. Lavoie and O. Rainbow, "A fast and portable realizer for text generation systems," in *Fifth Conference on Applied Natural Language Processing*, (Washington, DC, USA), pp. 265–268, Association for Computational Linguistics, 1997.

[65] A. Gatt and E. Reiter, "Simplenlg: A realisation engine for practical applications," in *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG '09, (USA), p. 90 – 93, Association for Computational Linguistics, 2009.

[66] S. W. Mcroy, S. Channarukul, and S. S. Ali, "An augmented template-based approach to text realization," *Nat. Lang. Eng.*, vol. 9, no. 4, p. 381 – 420, 2003.

[67] N. McIntyre and M. Lapata, "Learning to tell tales: A data-driven approach to story generation," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, (Suntec, Singapore), pp. 217–225, Association for Computational Linguistics, 2009.

[68] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, 2002.

[69] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27 of *NIPS'14*, (Cambridge, MA, USA), pp. 2672–2680, MIT Press, 2014.

[70] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, (Cambridge, MA, USA), p. 3104–3112, MIT Press, 2014.

[71] C. Hou, C. Zhou, K. Zhou, J. Sun, and S. Xuanyuan, "A survey of deep learning applied to story generation," in *Smart Computing and Communication*, (Cham), pp. 1–10, Springer International Publishing, 2019.

[72] K. Kumagai, I. Kobayashi, D. Mochihashi, H. Asoh, T. Nakamura, and T. Nagai, "Human-like natural language generation using Monte Carlo tree search," in *Proceedings of the INLG 2016 Workshop on Computational Creativity in Natural Language Generation*, (Edinburgh, UK), pp. 11–18, Association for Computational Linguistics, 2016.

[73] O. F. Zaidan and C. Callison-Burch, "Crowdsourcing translation: Professional quality from non-professionals," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Portland, Oregon, USA), pp. 1220–1229, Association for Computational Linguistics, 2011.

[74] X. Li, A. Taheri, L. Tu, and K. Gimpel, "Commonsense knowledge base completion," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1445–1455, Association for Computational Linguistics, 2016.

[75] I. Saito, K. Nishida, H. Asano, and J. Tomita, "Commonsense knowledge base completion and generation," in *Proceedings of the 22nd Conference on Computational Natural Language Learning*, (Brussels, Belgium), pp. 141–150, Association for Computational Linguistics, 2018.

[76] C. Malaviya, C. Bhagavatula, A. Bosselut, and Y. Choi, "Commonsense knowledge base completion with structural and semantic context," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, pp. 2925–2933, 2020.

[77] Y.-M. Hsieh, M.-H. Bai, S.-L. Huang, and K.-J. Chen, "Correcting chinese spelling errors with word lattice decoding," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 14, no. 4, 2015.

[78] K.-J. Chen, S.-L. Huang, Y.-Y. Shih, and Y.-J. Chen, "Extended-HowNet: A representational framework for concepts," in *Proceedings of OntoLex 2005 - Ontologies and Lexical Resources*, 2005.

[79] D. A. Cruse, *Lexical semantics*. Cambridge University Press, 1986.

[80] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

[81] Y. Niwa and Y. Nitta, "Co-occurrence vectors from corpora vs. distance vectors from dictionaries," in *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, p. 304–309, Association for Computational Linguistics, 1994.

[82] B. Daille, *Approche mixte pour l'extraction de terminologie : statistique lexicale et filtres linguistiques*. PhD thesis, Université Paris VII, 1994.

[83] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," *Proceedings,of the Biennial GSCL Conference 2009*, pp. 31–40, 2009.

[84] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in Neural Information Processing Systems*, vol. 27, pp. 2177–2185, Curran Associates, Inc., 2014.

[85] J. Caron, "Experiments with lsa scoring: Optimal rank and basis," in *PROC. OF SIAM COMPUTATIONAL INFORMATION RETRIEVAL WORKSHOP*, (USA), p. 157–169, Society for Industrial and Applied Mathematics, 2000.

[86] L. J. Bullinaria JA, "Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd," *Behavior Research Methods*, vol. 44, pp. 890–907, 2012.

[87] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 1480–1489, Association for Computational Linguistics, 2016.

[88] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[89] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37 of *ICML'15*, (Lille, France), p. 448–456, JMLR.org, 2015.

[90] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, (Berlin, Heidelberg), p. 282–293, Springer-Verlag, 2006.

[91] L.-W. Ku, Y.-T. Liang, and H.-H. Chen, "Opinion extraction, summarization and tracking in news and blog corpora," in *Proceedings of AAAI-CAAW-06, the Spring Symposia on Computational Approaches to Analyzing Weblogs*, 2006.

[92] S.-M. Wang and L.-W. Ku, "ANTUSD: A large Chinese sentiment dictionary," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, (Portorož, Slovenia), pp. 2697–2702, European Language Resources Association (ELRA), 2016.

[93] L.-C. Yu, L.-H. Lee, S. Hao, J. Wang, Y. He, J. Hu, K. R. Lai, and X. Zhang, "Building Chinese affective resources in valence-arousal dimensions," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 540–545, Association for Computational Linguistics, 2016.

[94] L. Xu, H. Lin, Y. Pan, H. Ren, and J. Chen, "Constructing the affective lexicon ontology," *Journal of the China Society for Scientific and Technical Information*, vol. 27, pp. 180–185, 2008.

[95] C.-R. Huang and S.-K. Hsieh, "Infrastructure for cross-lingual knowledge representation - towards multilingualism in linguistic studies," 2010.

[96] Y. Wu and W. Li, "Overview of the nlpcc-iccpol 2016 shared task: Chinese word similarity measurement," in *Natural Language Understanding and Intelligent Applications*, (Cham), pp. 828–839, Springer International Publishing, 2016.

[97] F. Hill, R. Reichart, and A. Korhonen, "SimLex-999: Evaluating semantic models with (genuine) similarity estimation," *Computational Linguistics*, vol. 41, no. 4, pp. 665–695, 2015.

[98] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and WordNet-based approaches," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (Boulder, Colorado), pp. 19–27, Association for Computational Linguistics, 2009.

[99] E. Bruni, N. K. Tran, and M. Baroni, "Multimodal distributional semantics," *J. Artif. Int. Res.*, vol. 49, no. 1, p. 1–47, 2014.

[100] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: Computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, (New York, NY, USA), p. 337–346, Association for Computing Machinery, 2011.

[101] P. Jin and Y. Wu, "SemEval-2012 task 4: Evaluating Chinese word similarity," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, (Montréal, Canada), Association for Computational Linguistics, 2012.

[102] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," *ACM Trans. Inf. Syst.*, vol. 20, no. 1, p. 116–131, 2002.

[103] C.-Y. Chen and W.-Y. Ma, "Word embedding evaluation datasets and wikipedia title embedding for chinese," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), 2018.

[104] Z. Yin and Y. Shen, "On the dimensionality of word embedding," 2018.

[105] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[106] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), 2018.

[107] Y. Wang, L. Cui, and Y. Zhang, "How can bert help lexical semantics tasks?," 2020.

[108] S. Gelly and Y. Wang, "Exploration exploitation in go: Uct for monte-carlo go," in *NIPS: Neural Information Processing Systems Conference On-line trading of Exploration and Exploitation Workshop*, (Canada), 2006.

[109] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2–3, p. 235–256, 2002.

Appendix

A. ConceptNet Relations

English ConceptNet relations used in daily life			
AtLocation	CapableOf	Causes	CausesDesire
CreatedBy	Desires	DistinctFrom	Entails
HasA	HasFirstSubevent	HasLastSubevent	HasProperty
HasPrerequisite	HasSubevent	IsA	InstanceOf
LocatedNear	MadeOf	MannerOf	MotivatedByGoal
NotCapableOf	NotDesires	NotHasProperty	PartOf
ReceivesAction	SymbolOf	UsedFor	
English ConceptNet relations not be used in daily life			
Antonym	dbpedia	DefinedAs	DerivedFrom
EtymologicallyDerivedFrom	EtymologicallyRelatedTo	ExternalURL	FormOf
HasContext	RelatedTo	SimilarTo	Synonym

Table A.1: English ConceptNet relations.

Table A.2: Chinese ConceptNet relations detailed information.

	Start concept	End concept	SurfaceText	Size
Antonym	x	x	...is an antonym of...	9280
AtLocation	EObj	place, Obj	...位在...裡/外/上面/下面、 可以在 ...找到...	20113
CapableOf	LO NO	A A, function	...會... ...會...	16794
Causes	S, V x E, V natural E, Obj, V E, V	S, V S, A negative E, S, V EObj, S S, V	因為...所以... ...會令人... ...會引發... ...會帶來... ...後會...	45623
CausesDesire	x S, V, time	A A	...會令人想要... ...的時候會想要...	21868
Desires	Obj	EObj, S, A	...想要...、...喜歡...	15300
HasFirstSubevent	E, S, A	active A	...時，要優先...	8326
HasProperty	x	S	...是...的	17663
HasSubevent	E, S, V, time place	A A	...的時候會... 在...會...	31689
HasA	Obj, place	Noun	...擁有...	3879
IsA	x	Noun	...是一種...	11131
MadeOf	NO Obj NO	Obj Obj Obj	...可用...製成、...可以做成... ...可用...組成 ...的原料是...	8965
MayUse	A	Obj	...的時候會用到...	10655
MotivatedByGoal	active A active A	EObj, S, active A EObj	想要...的時候會...、...是為了... 想要有...應該要...	27637
NotDesires	Obj E, S, V, time	x x	...懼怕/痛恨/厭惡... *...的時候，懼怕/痛恨/厭惡...	18809
PartOf	Noun	Noun	...是...的一小部分	2833
SymbolOf	x	x	...代表...	2520
Synonym	x	x	...is a synonym of...	15095

x : no restrictions
LO : living organisms
NO : non-living objects
E : event
A : action, subject is organism
V : verb, subject is organism or non-living object
S : status
Obj : living organisms and non-living objects
EObj : Event, living organisms and non-living objects
* : new created SurfaceText that not in Chinese ConceptNet

Table A.3: Chinese ConceptNet relations explanations and examples.

	Meaning	Chinese Example	English Example
Antonym	A is opposite of B	推 is an antonym of 拉	push is an antonym of pull
AtLocation	B is a place that something can in, out, on, under or beside it	主廚位在廚房裡	Chief in the kitchen
CapableOf	Organism can do something	朋友會幫忙	friends can help
	non-living objects have certain functions	烤箱會烘烤	oven can bake
Causes	B happens because A	因為相愛所以結婚	marry because love each other
	A makes someone feel B	失敗會令人難過	failure make someone feel sad
	A triggers B	鬥毆會引發受傷	fight cause injury
	A brings B. causality and non-causality	朋友會帶來歡樂、熱浪會帶來危險	friends bring happiness, heatwave will bring danger
	B will happen after A. causality and non-causality	吃飯後會食物中毒、吃飯後會讀書	get food poisoning after eating, study after eating
CausesDesire	A makes someone want to B	痛會令人想要哭	pain makes someone want to cry

	Someone wants to do B when A	讀書的時候會想要喝咖啡	want to drink coffee when studying
Desires	A wants/wants to B, A likes/likes to B	病人想要快樂	patient want to happy
HasFirstSubevent	Someone does B first when A	地震時，要優先保持冷靜	stay calm first when earthquake
HasProperty	B is a property of A	藥是苦的	medicine is bitter
HasSubevent	someone does B when A	下雨的時候會回家	go home when raining
	Someone does B in location A	在餐廳會吃午餐	eat lunch in restaurant
***HasA**	A has B	狗擁有耳朵	dog have ears
IsA	A is a B	袋子是一種容器	bag is a container
MadeOf	A is made from/of B, B is made into A	麵包可用麵粉製成	bread is made from flour
	A is made up of B	漢堡可由火腿組成	burger is made up of ham
	B is an ingredient of A	麵包的原料是牛奶	milk is an ingredient of bread
MayUse	Use B while doing A	睡覺的時候會用到毯子	Use blanket when sleeping
MotivatedByGoal	Someone does A in order to B	詢問是為了進步	ask in order to improve
	Someone should do B if he/she wants B	想要有健康應該要保養	adopt if you want children

NotDesires	A scares/hates/disgusts B	司機厭惡紅燈	driver hate red light
	*Someone scares/hates/disgusts B while doing A	*休息的時候，厭惡打擾	*scare ghost when watching horror film
PartOf	A is a part of B	眼睛是臉的一部分	eye is part of face
SymbolOf	A symbolically represents B	藍色代表憂鬱	red symbolically represents dangerous
Synonym	A and B are plesionyms	冷 is a synonym of 涼	cold is a synonym of cool

A: Start concept

B: End concept

underline: concept

*: new data(not in ConceptNet)

B. Data Cleaning of Relations in ConceptNet

Different relations have different methods to correct. Here are the detailed data cleaning information of each relation. The process of data cleaning can divided into modify, move (to other relations) and delete. Each relation may contain modification of SurfaceText and description of relation, which is more detailed than appendix A.3. [A] and [B] in SurfaceText refer to Start and End fields in ConceptNet. Text quoted by “ and ” is relation or SurfaceText.

AtLocation

Original SurfaceText

“你可以在 [B] 找到 [A]”

“[A] 在 [B] (裡、外、下)”

“[A] 有 [B]”

Modified SurfaceText

“可以在 [B] 找到 [A]”

“[A] 位在 [B] (裡、外、上面、下面)”

Description

The concept in End field is not always a physical location. It could be a virtual place, an organization, an object or a place that something can in, out, on, under or beside it.

Modify

Revert assertions when SurfaceText is “你可以在 [B] 找到 [A]” and “[A] 有 [B]”.

Move localizer in concept to SurfaceText to increase concept degree if

concept is separated, e.g., “[私房錢] 位在 [衣櫥 底下]” → “[私房錢] 位在 [衣櫥] 底下”. ([case-dough] is under [the wardrobe])

Move localizer in SurfaceText to concept if it is in vocabulary table [1] after moving, e.g., “[廚房] 在 [家] 裡” → “[廚房] 在 [家裡]” ([kitchen] is at [home]).

Move

Move to relation “HasA” when the SurfaceText is “[A] 有 [B]” or [B] is an abstract conception. For example, The church has windows. Although, windows are at the location church, it’s more precise that windows are part of the church or the church has the windows.

Delete

Delete concepts in End field if they’re not a location. Concept is a location if its category in Cilin [2] is “Cb” (C-時間與空間, Cb-空間). If specific term like [康是美], a store sells cosmetic and medicine, not in Chinese WordNet or Cilin. Check whether specific term in Start field in ConceptNet by relation “IsA” is a location or not.

CapableOf

Original SurfaceText

“[A] 能做的事情有 [B]”

“[A] 會 [B]”

Modified SurfaceText

“[A] 會 [B]”

[1] vocabulary table of word embedding

[2] HIT IR-Lab Tongyici Cilin (Extended), 哈工大信息检索研究室同义词词林扩展版

Description

・Organism can do something

・Non-living object can do something or have certain functions.

Move

・Move to relation "HasSubevent" when SurfaceText is "[A] 能做的事情有 [B]" and concept in Start field is verb, e.g., "[單戀] 能做的事情有 [等待]" (you can [wait] when [one-sided love]). Concept in Start field in relation "CapableOf" should be an organism which can take actions or an object has certain functions. Hence, verb in Start field would be inappropriate.

・Move to relation "Causes" when SurfaceText is "[verb] 會 [B]". For example, "[被罵] 會 [難過]" ([being_scolded] can [sad]). The SurfaceText should be "[A] 會令人 (make someone feel) [B]", not "[A] 會 (can) [B]".

Delete

Concept in Start field should be non-living object or can't act actively.

Causes

Original SurfaceText

"因為 [A] 所以 [B]"

"[A] 會讓你 [B]"

"[A] 可能會引起 [B]"

"[A] 可能會帶來 [B]"

"[A] 之後可能會發生的事情是 [B]"

Modified SurfaceText

“因為 [A] 所以 [B]”

“[A] 會令人 [B]”

“[A] 會引發 [B]”

“[A] 會帶來 [B]”

“[A] 後會 [B]”

Description

・“因為 [A] 所以 [B]”: causality between A and B. It’s a generalized SurfaceText of relation “Causes”. Most of the assertions with causal relation can use this. In contrast, it is not precise to describe the relation between A and B.

・“[A] 會令人 [B]”: A makes someone feel B.

・“[A] 會引發 [B]”: A brings about B, which B is usually something bad.

・“[A] 會帶來 [B]”: A causes an effect on B or A will bring in B

・“[A] 後會 [B]”: A causes an effect on B. After doing A, event or action B will be executed. One with causality and the other without causality. For example, “[吃] 後會 [飽]” and “[吃] 後會 [聊天]” (you will [feel stuffed] after [eating] and you will [chat] after [eating]).

Modify

Most of the assertions are not precise if SurfaceText is “因為 [A] 所以 [B]”. Modify these data to SurfaceText “[A] 會令人 [B]” and “[A] 後會 [B]”.

Move

Move to relation “HasSubevent” when SurfaceText is “因為 [A] 所以

[B]", e.g., "因為 [跳舞] 所以 [聽音樂]" → "[跳舞] 的時候，你會 [聽音樂]" (Because [dance], [listen to the music] → [listen to the music] when [dance]). Move to relation "MotivatedByGoal" if concept in Start field starts with "要", e.g., "因為 [要遺產] 所以 [結婚]" → "[結婚] 是為了 [要遺產]" (Because you [want heritage], you [marry] → you would [marry] because you [want heritage]).

CausesDesire

Original SurfaceText

"[A] 會讓你想要 [B]"

Modified SurfaceText

"[A] 會令人想要 [B]"

"[A] 的時候會想要 [B]"

Description

・"[A] 會令人想要 [B]": A makes someone wants to do B.

・"[A] 的時候會想要 [B]":

Someone wants to do B while doing A.

During a period of situation or event A, someone wants to do B.

Modify

"[A] 會讓你想要 [B]" → "[A] 的時候會想要 [B]" if A doesn't cause B to happen, e.g., "[孤單] 會令人想要 [睡覺]" → "[孤單] 的時候會想要 [睡覺]" ([lonely] makes someone wants to [sleep] → you want to [sleep] when you are [lonely]).

Desires

Original SurfaceText

“[A] 想要 [B]”

“[A] 喜歡 [B]”

“[A] 痛恨 [B]”

“[A] 懼怕 [B]”

“[A] 不想要 [B]”

Modified SurfaceText

“[A] 想要 [B]”

“[A] 喜歡 [B]”

Description

A wants or wants to B.

A likes or likes to B.

Move

・Move to relation “NotDesires” when SurfaceText contains 痛恨、懼怕、不想要.

・Move to relation “CausesDesire” if concept in Start field is verb, e.g., “[畢業] 想要 [工作]” → “[畢業] 的時候會想要 [工作]” ([graduate] want [a job] → you want [a job] when [graduate]).

Delete

Delete non-living objects in Start field.

HasFirstSubevent

Original SurfaceText

“[A] 的時候，首先要 [B]”

Modified SurfaceText

“[A] 時，要優先 [B]”

Description

Someone does B first when A.

Move

Some of the assertions in relation "HasFirstSubevent" should be relation "HasPrerequisite". For example, "[煮飯] 時，首先要 [買食材]" → "[煮飯] 前，首先要 [買食材]" (you will [buy ingredients] first while [cooking] → you will [buy ingredients] before [cooking]). The meaning of relation "HasFirstSubevent" is you will do B first while doing A, not before. There is no relation "HasPrerequisite", which means doing something before something, in Chinese ConceptNet. Move these data to relation "MotivatedByGoal" and revert assertions, such as "[買食物] 是為了 [煮飯]" ([buy ingredients] in order to [cook]). It means you need to buy ingredients first to achieve the goal of cooking.

HasProperty

Description

B is a property of A.

Move

There are two meanings about SurfaceText "[A] 是 [B] 的" in Chinese. One is A has property B and another is A is belonged to B, e.g., "[小孩] 是 [聰明] 的" and "[書] 是 [他] 的" ([child] HasProperty [smart] and [the book] HasProperty [him]). In relation "HasProperty", the latter is wrong. B should be an adjective which can describe A, not an organism who can own something. Move to relation "HasA" if B is an organism.

HasSubevent

Original SurfaceText

"[A] 的時候，你會 [B]"

"在 [A]，你會 [B]"

"[A] 可能代表 [B]"

Modified SurfaceText

"[A] 的時候會 [B]"

"在 [A] 會 [B]"

Description

・"[A] 的時候會 [B]": Do B while A

・"在 [A] 會 [B]": In location A, you will do B.

Move

・Move to relation "SymbolOf" when SurfaceText is "[A] 可能代表 [B]".

・Move to relation "HasFirstSubevent" if concept in End field contains "先".

HasA

Original SurfaceText

There is no relation "HasA" in earlier version of Chinese ConceptNet.

Modified SurfaceText

"[A] 擁有 [B]"

MadeOf

Original SurfaceText

"[A] 可以用 [B] 製成"

"[A] 由 [B] 組成"

"[A] 的原料是 [B]"

"[A] 可以拿來做成 [B]"

Modified SurfaceText

"[A] 可用 [B] 製成"

"[A] 可用 [B] 組成"

"[A] 的原料是 [B]"

"[B] 可以做成 [A]"

Description

・"[A] 可用 [B] 製成": need to be processed

・"[A] 可用 [B] 組成": combination of different components

・"[A] 的原料是 [B]": food ingredients

・"[B] 可以做成 [A]": need to be processed

Modify

・Revert assertions when SurfaceText is "[B] 可以拿來做成 [A]".

・Modify the SurfaceText to be more precise. Some concepts are physical combination, and some concepts need to be processed.

MotivatedByGoal

Original SurfaceText

"[A] 是為了 [B]"

"當你想要 [B] 的時候你可能會 [A]"

"想要有 [B] 應該要 [A]"

"你會 [A] 因為你 [B]"

"[A] 的時候會想要 [B]"

Modified SurfaceText

“[A] 是為了 [B]”

“想要 [B] 的時候會 [A]”

“想要有 [B] 應該要 [A]”

Description

・“[A] 是為了 [B]”: Someone does A in order to B

・“想要 [B] 的時候會 [A]”: Someone does A in order to B

・“想要有 [B] 應該要 [A]”: Someone should do B if he/she wants B

Modify

・Revert assertions when SurfaceText is “想要 [B] 的時候會 [A]” and “想要有 [B] 應該要 [A]”.

・The SurfaceText “想要 [B] 的時候會 [A]” contains two meanings. The first one is someone does A because they want result B, e.g., “想要 [讀書] 的時候會 [翻開書]” (You’ll [open the book] when you want to [read]). The second one is someone does A because they don’t want result B, e.g., 想要 [睡覺] 的時候會 [買咖啡] (You’ll [buy a coffee] when you want to [sleep]). B is usually a physiological needs like sleeping, crying, sneezing or yawning If B is going to happen, you will take actions to deal with or prevent it. However, the meaning of relation “MotivatedByGoal” is doing A to achieve the goal B. Therefore, the second one is inaccurate, and one of the concept pair need to be modified.

Move

・Move to relation “CausesDesire” when SurfaceText is “A 的時候會想要 B”.

・Move to relation “HasSubevent” when SurfaceText is “你會 A 因為你 B”, e.g., “你會 [喝飲料] 因為你 [口渴]” → “[口渴] 的時候會 [喝飲料]”.

NotDesires

Original SurfaceText

“[A] 懼怕/痛恨/厭惡 [B]”

Modified SurfaceText

“[A] 懼怕/痛恨/厭惡 [B]”

“[A] 的時候，懼怕/痛恨/厭惡 [B]”

Description

・“[A] 懼怕、痛恨、厭惡 [B]”: A scares/hates/disgusts B

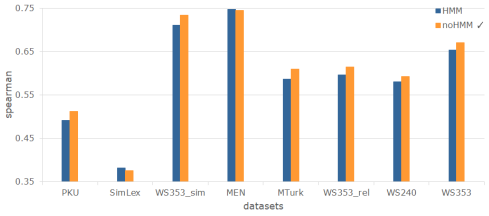
・“[A] 的時候，懼怕、痛恨、厭惡 [B]”: Someone scares/hates/disgusts B while doing A

Modify

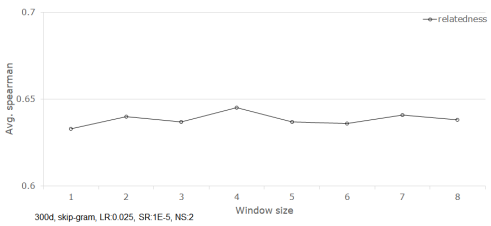
Create a new SurfaceText “A 的時候，懼怕、痛恨、厭惡 B” when the concept in Start field is a verb or a period of time.

C. Word Embedding Experiments

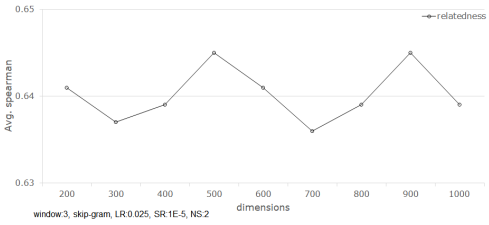
Prediction-based



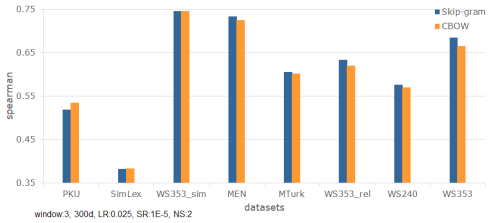
(a) Discover new words with or without HMM.



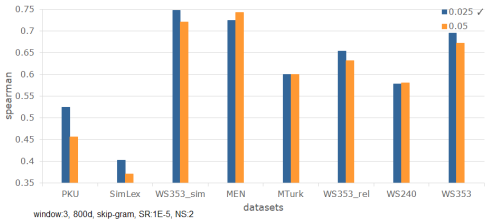
(b) Window size.



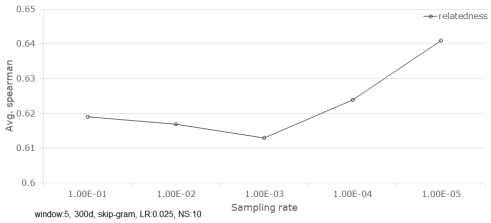
(c) Dimensions.



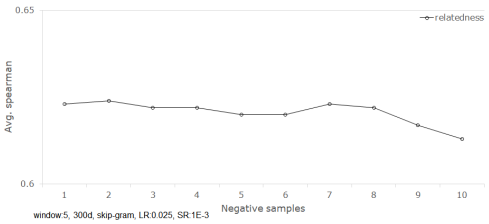
(d) Skip-gram and CBOW.



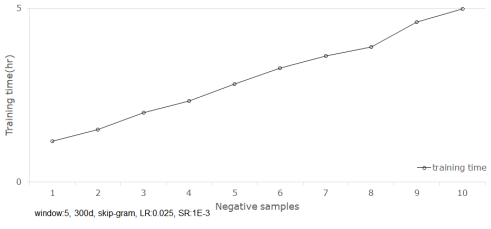
(e) Learning rate.



(f) Sampling rate.



(g) Negative samples.



(h) Training time in different negative samples.

Word embedding	Dim	Similarity			Avg.	Combined
		PKU	SimLex	WS353_sim		WS353
Our WE model_count	1500	.524	.395	.749	.556	.722
Our WE model_prediction	500	.554	.403	.741	.566	.669
CKIP_Glove	300	.372	.326	.577	.425	.516
CKIP_word2vec	300	.438	.346	.698	.494	.657
fastText_wiki	300	.342	.104	.393	.280	.242
fastText_cc	300	.493	.380	.616	.496	.608
Numberbatch	300	.446	.420	.768	.545	.700
BERT	768	.253	.218	.484	.318	.400

Dim: Dimension

Table A.4: Compare to other pre-trained word embeddings on similarity datasets.

Word embedding	Vocab size	Similarity			Combined	Avg.(%)
		PKU	SimLex	WS353_sim	WS353	
Our model	249,571	52	76	14	23	8.1
CKIP	480,551	59	34	10	19	6.3
fastText_wiki	50,184	109	73	32	48	12.8
fastText_cc	307,441	31	26	8	16	4
Numberbatch	300	.446	.420	.768	.545	.700

Table A.5: Compare to other pre-trained word embeddings (OOV).

Hyperparameter	best settings
Frequency weighting	SPPMI_k10
Window size	2
Dimensions	1500
Remove first k dimensions	26
Weighting exponent p	0.7
Discover new words	no

(a) Count-based.

Hyperparameter	best settings
Window size	2
Dimensions	500
Model	CBOW
Learning rate (LR)	0.025
Sampling rate (SR)	0.00001
Negative samples (NS)	2
Discover new words	no

(b) Prediction-based.

Table A.6: Best hyperparameter settings in similarity task.

D. Recombine words and simplify POS

精簡詞類	簡化標記	對應的CKIP詞類標記[1]	
A	A	A	/*非謂形容詞*/
C	Caa	Caa	/*對等連接詞，如：和、跟*/
POST	Cab	Cab	/*連接詞，如：等等*/
POST	Cba	Cbab	/*連接詞，如：的話*/
C	Cbb	Cbaa, Cbba, Cbbb, Cbca, Cbcb	/*關聯連接詞*/
ADV	Da	*Daa*	/*數量副詞*/
ADV	Dfa	Dfa	/*動詞前程度副詞*/
ADV	Dfb	Dfb	/*動詞後程度副詞*/
ASP	Di	Di	/*時態標記*/
ADV	Dk	Dk	/*句副詞*/
ADV	D	*Dab*, Dbaa, Dbab, Dbb, Dbc, Dc, Dd, Dg, Dh, Dj	/*副詞*/
N	Na	Naa, Nab, Nac, Nad, Naea, Naeb	/*普通名詞*/
N	Nb	Nba, Nbc	/*專有名稱*/
N	Nc	Nca, Ncb, Ncc, Nce	/*地方詞*/
N	Ncd	Ncda, Ncdb	/*位置詞*/
N	Nd	Ndaa, Ndab, Ndc, Ndd	/*時間詞*/
DET	Neu	*Neu*	/*數詞定詞*/.
DET	Nes	*Nes*	/*特指定詞*/
DET	Nep	*Nep*	/*指代定詞*/
DET	Neqa	*Neqa*	/*數量定詞*/
POST	Neqb	*Neqb*	/*後置數量定詞*/
M	Nf	Nfa, Nfb, Nfc, Nfd, Nfe, Nfg, Nfh, Nfi	/*量詞*/
POST	Ng	Ng	/*後置詞*/
N	Nh	Nhaa, Nhab, Nhac, Nhb, Nhc	/*代名詞*/
Nv	Nv	Nv1,Nv2,Nv3,Nv4	/*名物化動詞*/
T	I	I	/*感嘆詞*/
P	P	P*	/*介詞*/
T	T	Ta, Tb, Tc, Td	/*語助詞*/
Vi	VA	VA11,12,13,VA3,VA4	/*動作不及物動詞*/
Vt	VAC	VA2	/*動作使動動詞*/
Vi	VB	VB11,12,VB2	/*動作類及物動詞*/
Vt	VC	VC2, VC31,32,33	/*動作及物動詞*/
Vt	VCL	VC1	/*動作接地方賓語動詞*/
Vt	VD	VD1, VD2	/*雙賓動詞*/
Vt	VE	VE11, VE12, VE2	/*動作句賓動詞*/
Vt	VF	VF1, VF2	/*動作謂賓動詞*/
Vt	VG	VG1, VG2	/*分類動詞*/
Vi	VH	VH11,12,13,14,15,17,VH21	/*狀態不及物動詞*/
Vt	VHC	VH16, VH22	/*狀態使動動詞/
Vi	VI	VI1,2,3	/*狀態類及物動詞*/
Vt	VJ	VJ1,2,3	/*狀態及物動詞*/
Vt	VK	VK1,2	/*狀態句賓動詞*/
Vt	VL	VL1,2,3,4	/*狀態謂賓動詞*/
Vt	V_2	V_2	/*有*/
T	DE	/*的, 之, 得, 地*/	
Vt	SHI	/*是*/	
FW	FW	/*外文標記*/	
COLONCATEGORY			/* 冒號 */

Figure 2: POS tags in CKIP tagger.

The POS tags are shown in Figure 2 [3]. We have to recombine words separated by CKIP tagger because we sue Jieba as our word segmentation tool. The number

[3] http://ckipsvr.iis.sinica.edu.tw/

Caa, Cab, Cba, Cbb	C(連接詞)
D, Da, Dfa, Dfb, Di, Dk, DM	D (副詞)
Na	N (體詞)
VA, VAC, VB, VC, VD, VI	V (述詞)
VCL, VE, VF, VK, VHC	conditional V1 (條件動詞 1)
VH, VJ, VL	conditional V2 (條件動詞 2)
	C
CR	Neu, Nes, Nep, Neqa, Neqb, Nf, Ng
	I, P, T, VG, V_2, DE, SHI, FW

CR: can't be replaced

Table A.7: POS simplification.

of segmented words by Jieba is fewer than CKIP tagger, but CKIP tagger performs better in POS tagging. Because we don't need detailed POS, we simplify the POS tags in table A.7. Conditional V1 and V2 are verb in some conditions. We add a new tag CR (can't be replaced). It's not an official POS, it's for our research purpose. Some parts-of-speech have little semantic meaning, and concepts with these parts-of-speech barely appeared in ConceptNet. Therefore, we don't replace them.

We define some rules to combine separated parts-of-speech. The symbol P represents word POS. The symbols in [] are combined results. The left side of → is original POS, and the right side of → is its POS after processing.

Single part-of-speech

[Nv → V]

Nv is nominalized verb. Although Nv is noun in CKIP's POS technical report [4]. We still classify Nv to verb after examinations.

[P]

If P not in CR or return CR if P in CR.

[4] https://ckip.iis.sinica.edu.tw/CKIP/paper/TR9305.pdf

Muitiple parts-of-speech

we list the order of POS combinations.

- [Nb, FW → CR]

  Return Nb or CR if one of them in parts-of-speech. Because lots of the proper noun can't be detected, it may be a proper noun if Nb in the parts-of-speech. For example, 陳 明志 (Nb+V) → 陳明志 (Nb). FW is the tag of foreign languages. Most of them are unknown words. We tag it as CR.

- If all of the parts-of-speech are the same.

  [Nv → V]

  [P]

  Return P if not in CR. For example, 衝 下去 (VCL+VCL) → 衝下去 (VCL), 紙 風車 (N+N) → 紙風車 (N).

- [VH+Nc → Nc]

  For example, 新 家 (VH+Nc) → 新家 (Nc), 偏遠 地區 (VH+Nc) → 偏遠地區 (Nc).

- [Nv+VJ, Nv+VL, VH+N, VHC+N → N]

  For example, 檢驗 費 (Nv+VJ) → 檢驗費 (N), 失蹤 人口 (VH+Na) → 失蹤人口 (N), 壞 習慣 (VHC+Na) → 壞習慣 (N).

- [[VCL, VE, VF, VK, VHC]+P → V+P]

  If VCL, VE, VF, VK, VHC combine with other parts-of-speech, change their POS to V. For example, 供 油 (VF+Na) → 供 油 (V+Na), 顧 大局 (VK+N) → 顧 大局 (V+N).

- [[VH, VJ, VL]+Vx → V+Vx]

  Vx is a set of verbs with includes all of the verbs in Figure 2. For example,

- [...+ N + ...+ N + ...] → [...+ N + ...]

  Merge adjacent nouns to one noun except Nc. For example, 國家 衛生 研究院 (N+N+Nc) → 國家衛生 研究院 (N+Nc), 年 初二 (Nd+Neu) → 年初二 (N)

- [VH, VL, VJ, V, Nc, N, Nv→V, CR]

  After combining these parts-of-speech, some of the words are still fragment. Return P if P in above ordered list. Because VH, VL, VJ are more detailed than V, we put it before V. It's the same in Nc and N.

The parts-of-speech after combinations are shown in table A.8.

A (非謂形容詞)
D (副詞)
N (體詞)
Nb (專有名詞)
Nc (地方詞)
Ncd (方位詞)
Nd (時間詞)
Nh (代名詞)
V (述詞)
VH (狀態不及物動詞)
VJ (狀態及物動詞)
VL (狀態謂賓動詞)
CR (Can't be Replaced)

Table A.8: POS after combining.

E. Coherence model

Max word num	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
10	.489	.765	.437	.799	.435	.798
15	.486	.766	.436	.798	.437	.797

Table A.9: The max number of words in a sentence.

Bidirectional merge mode	Train		Validation		Test	
	loss	acc	loss	acc	loss	acc
Summation	.178	.932	.140	.949	.134	.952
Multiplication	.176	.934	.133	.951	.128	.955
Concatenation	.153	.944	.127	.955	.122	.957
Average	.180	.931	.140	.948	.136	.951

Table A.10: Bidirectional merge mode.

F. Templates

Sentence Templates

Table A.11: Sentence templates.

	AtLocation	CapableOf	Causes	CausesDesire
AtLocation	可以在 [餐廳] 找到 [蟑螂] 和 [筷子] 可以在 [餐廳] 和 [桌子] 找到 [蟑螂]	x		
CapableOf	[天堂] 的 [上帝] 會 [拯救人類]	[小狗] 會 [奔跑] 和 [玩樂]		
Causes	[網咖] 的 [電腦] 會令人 [上癮]	[羊][喝水] 後會 [涼爽]	因為 [疲倦] 所以 [睡覺] 後會 [有精神] [下班] 後 [吃東西] 會 [胖] 因為 [輻合作用] 所引發的 [龍捲風] 會導致 [風災]	
CausesDesire	[網咖] 的 [電腦] 會令人想要 [使用]	[無聊][看電影] 時會想要 [進食]	因為 [吹風] 而 [冷] 時會想要 [打人] [中暑] 後 [頭暈] 時會想要 [休息]	[狗] 會令人想要 [疼牠]、[跟他玩]
Desires	[捷運] 的 [上班族] 想要 [颱風假]	[老公] 喜歡會 [打掃] 的 [妻子]	[上班族] 想要會令人 [輕鬆] 的 [旅遊] [老公][跑步] 後會想要 [睡覺]	x
NotDesires	[捷運] 的 [上班族] 痛恨 [加班]	[老公] 厭惡會 [打掃] 的 [妻子] [老公][騎車] 時懼怕 [下雨]	[老公] 厭惡會令人 [傷腦筋] 的 [老婆] [騎車] 時懼怕 [跌倒] 會令人 [受傷] [老公] 厭惡 [運動] 後 [睡覺]	[他] 痛恨 [退學]，會令人想要 [哭] [肚子痛] 時厭惡 [上學]，會令人想要 [睡覺]
HasFirstSubevent	x	[一個人][看電影] 時要優先 [付錢]	[下班] 時要優先 [吃飯] 才會令人 [不煩悶] [下班] 時要優先 [吃飯] 再 [洗澡]	x
HasProperty	[巧克力展] 有 [咖啡色] 的 [巧克力]	[善良] 的 [主人] 會 [餵食]	[夢幻] 的 [愛情] 會令人 [失落] [暢快] 的 [喝酒] 後會 [呼呼大睡]	[開心] 的 [跳舞] 時會想要 [拉筋]
HasSubevent	在 [倉庫] 的 [廚房][煮開水]	[大學生] 會在 [白天][跑步] 在 [座位][看書] 會令人 [有內涵]	[肚子餓] 時 [吃麵] 會令人 [不餓] [大學生][讀書] 時會 [打開書] 因為 [感冒] 而 [頭痛] 時會 [睡覺] [看書] 後 [玩電腦] 時會 [聽音樂]	[上課][打瞌睡] 時會想要 [洗把臉] 在 [中秋節][約會] 時會想要 [逛夜市]
IsA	[學校] 的 [電腦] 是一種 [便利發明]	[男人] 是會 [煮菜] 的 [生物]	[戀愛] 是會令人 [幸福] 的 [魔法]	[戀愛] 是會令人想要 [佔有] 的 [魔法]
MadeOf	[餐廳] 的 [魚] 可以製成 [炒飯] [冰箱] 的 [便當] 可用 [米飯] 組成	[鋼鐵] 做成的 [菜刀] 可以 [切菜]	[植物] 製成的 [食物] 會令人 [飽]	[水] 組成的 [小河] 會令人想要 [游泳]
MotivatedByGoal	x	[大人][上班] 是為了 [養家]	為了 [休息] 而 [抽煙] 會令人 [肺病]	[休息] 時會想要 [看電影] 來 [打發時間] [看電影] 是為了 [打發時間]，會令人想要 [睡覺]
PartOf	x	x	x	x
SymbolOf	x	[狗][流口水] 表示 [肚子餓]	x	x
MayUse	x	[綿羊][喝水] 時會用到 [水龍頭]	用 [水龍頭][喝水] 後會 [跑廁所]	用 [手][按摩] 時會想要 [跑廁所]
HasA	[台北] 的 [路人] 擁有 [手機]	[我] 的 [嘴巴] 可以 [吃東西]	x	[全家] 的 [食物] 會令人想要 [吃]

blanks are duplicate of other templates

x: no this template

	Desires	NotDesires	HasFirstSubevent	HasProperty
AtLocation			x	
CapableOf				x
Causes				
CausesDesire	x		x	
Desires	[蝴蝶] 喜歡 [草原] 和 [花蜜]			
NotDesires	[蝴蝶] 喜歡 [飛舞] 但害怕 [蜘蛛]	[學生] 和 [弟弟] 厭惡 [上學] [學生] 厭惡 [上學] 和 [作業]	x	
HasFirstSubevent	[孩子] 想要 [吃飯] 時要優先 [準備碗筷]	x	x	x
HasProperty	[學校] 想要 [認真] 的 [老師]	[小狗] 不想要 [幸福] 的 [巧克力]	x	[家庭] 是 [溫馨]、[幸福] 的
HasSubevent	[哥哥] 想要 [讀書] 時會 [開冷氣]	[哥哥] 不想要 [唸書] 時 [喝咖啡] [上班] 時 [上網] 懼怕 [網路斷線] 在 [廚房][做菜] 時懼怕 [燙傷]	[煮飯] 時 [唱歌] 要優先 [喝水] [唱歌] 時要優先 [挑歌] 再 [饒舌] 在 [中午][吃飯] 時要優先 [洗手]	在 [花花綠綠] 的 [公園][踢足球] [追女生] 時會 [勇敢] 的 [告白]
IsA	[倉鼠] 是喜歡 [滾輪] 的 [哺乳類]	[玩具] 是不想要 [壞掉] 的 [娛樂物品]	x	[餅乾] 是 [可口] 的 [垃圾食物]
MadeOf	[人] 喜歡 [電路板] 組成的 [電腦]	[殭屍] 懼怕 [鞭炮] 做成的 [武器]	x	[奶油] 做成的 [餅乾] 是 [熱量高] 的
MotivatedByGoal	[學生] 想要為了 [將來] 而 [努力]	x	[看牙醫] 時要優先 [抹牙膏][刷牙] [睡覺] 時要優先 [拿遙控器] 是為了 [關燈] [拿鏡子] 化妝] 時要優先 [上粉底]	[乖乖聽話] 是為了 [誠] 的 [餅乾] [健康] 的 [吃早餐] 是為了 [充飢]
PartOf	x	x	x	[西餐] 的 [麵包] 是 [軟] 的
SymbolOf	x	x	x	x
MayUse	x	x	使用 [擴音器][講電話] 時要優先 [撥號碼]	[生病] 時會用到 [苦] 的 [中藥]
HasA	[獅子] 擁有 [利牙]，喜歡 [大開口]	擁有 [利牙] 的 [獅子] 懼怕 [火焰]	x	擁有 [傘] 的 [路人] 是 [冷淡的] [路人] 擁有 [便宜] 的 [手機]

	HasSubevent	IsA	MadeOf	MotivatedByGoal
AtLocation				x
CapableOf				
Causes				
CausesDesire				
Desires				
NotDesires				x
HasFirstSubevent		x	x	
HasProperty				
HasSubevent	[疲勞] 時會 [洗澡] 和 [睡覺] 在 [公司] 時會 [上網] 和 [偷懶] [想吃東西] 時會 [騎車][出門] 在 [公司][工作] 時會 [打混]	x	x	
IsA	x	x		x
MadeOf	x	[麵包] 是 [中筋麵粉] 製成的 [乾糧] [麵包] 是一種可以做成 [蟹堡] 的 [乾糧]	[窗戶] 可由 [鋼] 和 [木條] 製成	
MotivatedByGoal	[難過] 時會 [大吃] 來 [發洩] 在 [辦公室][工作] 是為了 [家計]	x	[逛光華商場] 是為了 [主機] 組成的 [電腦]	x
PartOf	x	[舌頭] 是 [身體] 的一部份 [黑幫分子] 的 [囚犯] 是 [受刑人]	x	x
SymbolOf	[運動] 時 [疲勞] 代表 [需要休息]	x	x	為了 [脫離單身] 而 [告白] 代表 [友誼結束]
MayUse	[約會][喝咖啡] 時會用到 [杯子] 在 [咖啡店][喝咖啡] 時會用到 [杯子]	[郊遊] 時會用到 [冰淇淋]	[上學] 時會用到 [皮] 製成的 [包包]	x
HasA	x	[女人] 是有 [頭髮] 的 [雌性動物]	[主人] 擁有 [印鈔機] 製成的 [錢]	x

	PartOf	SymbolOf	MayUse	HasA
AtLocation	x	x	x	
CapableOf	x			
Causes	x	x		x
CausesDesire	x	x		
Desires	x	x	x	
NotDesires	x	x	x	
HasFirstSubevent	x	x		x
HasProperty				
HasSubevent	x			x
IsA		x		
MadeOf	x			
MotivatedByGoal	x		x	x
PartOf	x			x
SymbolOf	[商業] 的 [電影] 是 [娛樂] 的一部分	x	x	
MayUse	[作弊] 時會用到 [臉部] 的 [眼睛]	x	[上學] 和 [旅行] 時會用到 [交通工具] [出國] 時會用到 [信用卡] 和 [地圖]	
HasA	x	[人] 擁有代表 [情感] 的 [愛心]	[男生] 有 [指甲]，[抓癢] 時會用到	[鳥] 擁有 [頭] 和 [毛]

Paragraph Templates

Table A.15: Paragraph templates

AtLocation	HasProperty
CapableOf	CapableOf
Desires	Desires
IsA	Causes
[收容所] 有 [可愛] 的 [狗]	
[狗] 會 [奔跑] 和 [玩樂]	
想要 [有人收養] 和 [有好的主人]	
是一種 [貼心的動物]，會令人 [開心]	

AtLocation	HasProperty	
Causes	IsA	
CausesDesire	CausesDesire	
HasSubevent	HasProperty	MotivatedByGoal
[公園] 有 [可愛] 的 [貓]		
是會令人 [開心] 的 [動物]		
會令人想要 [寵愛]、[抱抱]		
[抱抱] 時會 [甜蜜] 的 [親親] 來 [示好]		

CapableOf	MotivatedByGoal	
CausesDesire	HasProperty	
HasSubevent	MotivatedByGoal	HasProperty
HasProperty	Causes	
[老公][工作] 是為了 [有錢]		
[煩悶] 的 [工作] 會令人想要 [逃避]		
[有錢] 的時候會 [買房子]，是為了 [快樂] 的 [生活]		
但 [痛苦] 的 [房貸] 會帶來 [壓力]		

AtLocation	CapableOf	
HasFirstSubevent	HasSubevent	
HasProperty	MotivatedByGoal	Causes
Causes	CausesDesire	MotivatedByGoal
[朋友] 在 [公司][工作]		
[工作] 時首先要 [吃早餐] 再 [認真]		
[健康] 的 [吃早餐] 是為了 [充飢] 會帶來 [活力]		
[工作] 後 [疲倦] 時會想要 [睡覺] 來 [休息]		

Desires		
Causes	HasProperty	
NotDesires	Causes	
CapableOf	HasSubevent	
[老公] 喜歡 [美女]		
[美麗] 的 [美女] 會令人 [歡愉]		
[老公] 不喜歡會令人 [傷腦筋] 的 [老婆]		
[她][抓狂] 後會 [打人]		

CapableOf	HasFirstSubevent	MotivatedByGoal
HasProperty	MotivatedByGoal	
NotDesires	Causes	
HasSubevent	MotivatedByGoal	
[學生][上課] 時要優先 [拿筆][作筆記]		
[辛苦] 的 [讀書] 是為了 [將來]		
[讀書] 時厭惡 [懶惰]，會令人 [落榜]		
[煩悶] 時會 [聽音樂] 來 [冷靜]		

NotDesires HasProperty

Causes

Causes HasSubevent MotivatedByGoal HasProperty

Causes HasProperty

[朋友] 厭惡 [辛苦] 的 [工作]

[出錯] 會令人 [自責]

[工作] 後 [有錢] 時會 [買房子]，是為了 [快樂] 的 [生活]

但 [痛苦] 的 [房貸] 會帶來 [壓力]

CapableOf HasProperty

HasFirstSubevent HasSubevent

NotDesires Causes

Causes

[賢慧] 的 [女友] 會 [煮菜]

[煮菜] 時要優先 [洗手] 再 [試味道]

[放油] 時懼怕 [噴濺] 會令人 [受傷]

[美味的食物] 會令人 [食指大動]