CKIP at the NTCIR-13 STC-2 Task

Wei-Yun Ma
Academia Sinica
Taipei, Taiwan
886-2-27883799
ma@iis.sinica.edu.tw

Chien-Hui Tseng
National Taiwan University
Taipei, Taiwan
886-2-27883799
r05725004@ntu.edu.tw

Yu-Sheng Li
National Taiwan University
Taipei, Taiwan
886-2-27883799
b03902086@ntu.edu.tw

ABSTRACT
In recent years, LSTM-based sequence-to-sequence model have been applied successfully in many fields, including short text conversation and machine translation. The inputs and outputs of the models are usually word sequences. However, for a fixed-size training corpus, a word sequence or even part of it is unlikely to repeat many times, thus in natural, data sparseness problem could be an obstacle for training of sequence-to-sequence model. To address this issue, through this task, we propose the idea of using LSTM with concept sequence. That is, given input word sequence, we first predict the concept for each word of the word sequence and thus form a concept sequence as the input of the LSTM model. At training phase, the output remains the form of word sequence. So during testing phase, given a generated concept sequence, LSTM model is able to directly output the corresponding response in a form of word sequence. Although our results are not among top systems in this task, the experimental results still show the potential of this idea through the comparison among our submitted runs.

Keywords
artificial intelligence, dialogue systems, sequence-to-sequence, encoder-decoder, deep learning, recurrent neural network, long-short-term-memory, natural language processing

Team Name
ckip

Subtasks
STC-2

1. INTRODUCTION
NTCIR-13 Short Text Conversation (STC) task provided a platform for both retrieval-based method [1] and generation-based method [2][3][4]. Retrieval-based methods do not generate any new text and just select a response from the dataset that STC provided. On the other hand, generation-based methods do not rely on pre-defined responses but instead, aim to generate new responses from scratch. In general, generation-based methods are harder to train and the generated responses have more grammatical errors, but in natural they have more potential to handle unseen cases for which no appropriate predefined response exists. Therefore, for STC task this year, we only attended track of generation-based method and submitted four different runs.

Generation-based methods typically treat response generation as a problem of machine translation, but instead of generating translations from one language to another, they generate responses by given posts in the same language. [2] uses conventional phrase-based statistical machine translation to implement the strategy. In contrast, in recent years, STC using RNN-based models emerge [3][4] and gradually cause more attentions especially when many researches have shown the success of MT using RNN-based models. [5][6] are of the pioneering work in this thread. They propose employing a neural encoder-decoder model. A post is first summarized as a vector representation by a RNN-based encoder, and the vector representation is then fed to a RNN-based decoder to generate the corresponding response. Their experimental results show that the performance outperforms the traditional retrieval-based and translation-based methods.

Neural encoder-decoder model are also called sequence-to-sequence model, in which its inputs and outputs are usually word sequences in either STC or MT. In fact, for a fixed-size STC corpus, a word sequence or even part of it is unlikely to repeat many times, thus in natural, data sparseness problem could be an obstacle for training of sequence-to-sequence model.

To address this issue, in this paper, we, team CKIP, propose a ConceptSequence-to-WordSequence model (CS-to-WS model) and regard it as the major submitted run for this STC task. The concept list is collected from an entity-relation common-sense representation system, named Extend-HowNet (Ehownet). Given input word sequence, we first predict the concept for each word of the word sequence and thus form a concept sequence as the input of the LSTM model[7]. At training phase, the output remains the form of word sequence. So during testing phase, given a generated concept sequence, LSTM model is able to directly output the corresponding response in a form of word sequence.

2. BACKGROUND
2.1 LSTM-based Model
The Recurrent Neural Network is a natural generalization of feedforward neural networks to sequences, which allows for time-delayed directed cycles between units. Following [5]'s notation, the LSTM we used is described below. Given a sequence of inputs $(x_1, \ldots, x_T)$, the goal of the RNN model is to generate a sequence of outputs $y_1, \ldots, y_{T'}$, which conditional probability $P(y_1, \ldots, y_{T'} \mid x_1, \ldots, x_T)$ is estimated to be highest among all possible sequences of outputs.

The LSTM encoder computes this conditional probability by first obtaining the fixed dimensional representation v of the input sequence $(x_1, \ldots, x_T)$ given by the last hidden state of the LSTM, and then feed the v to a LSTM decoder as input. So the goal of the decoder is to estimate $P(y_1, \ldots, y_{T'} \mid x_1, \ldots, x_T)$ using (1)

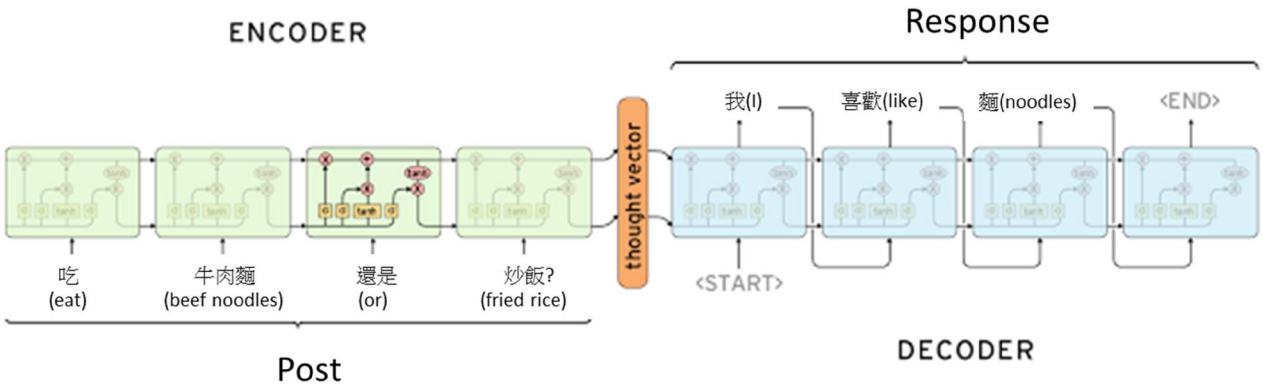

Figure 1. An LSTM model with Post- " 吃牛肉麵還是炒飯?(eat beef noodles or fried rice?)" as its input and response"我喜歡麵 (I like noodles)"> as its output. Diagram is modified from the original one at Google Research Blog.

$$P(y_1,\ldots,y_{T'} \mid x_1,\ldots,x_T) = \prod_{t=1}^{T'} P(y_t \mid v, y_1,\ldots,y_{T'}) \qquad (1)$$

$$P(y_t \mid v, y_1,\ldots,y_{T'}) = g(Wh_t)$$

g(.) is a softmax activation function, and h_t is the hidden state of decoder at time t calculated by

$$h_t = f(y_{t-1}, h_{t-1}, v)$$

f(.) is a LSTM unit with parameters for input, forget and output gates. In training, we add a special end-of-sentence symbol "END" to the output sequence, which enables the model to define a distribution over generated sequences of all possible lengths. Take fig1 as an example, in the training phase, given post-response pair – <" 吃牛肉麵還是炒飯?(eat beef noodles or fried rice?)", "我喜歡麵(I like noodles)">, one can calculate P(我喜歡麵 END|吃牛肉麵還是炒飯?) by (1) and use statistical gradient decent to adjust every parameter of both LSTM encoder and decoder.

2.2 HowNet and Extend-Hownet

Given input word sequence, we first predict the concept for each word of the word sequence and thus form a sense sequence as the input of the LSTM model. The concept list is collected from an entity-relation common-sense representation system, named Extend-HowNet. At training phase, the output remains the form of word sequence. So during testing phase, given a generated concept sequence, LSTM model is able to directly output the corresponding response in a form of word sequence. In this section, we briefly introduce HowNet [8] and Extend-Hownet [9][10].

Extend-HowNet[1] is an extended version of HowNet, which is an common-sense knowledge base unveiling the inter-conceptual relations and inter-attribute relations of concepts conveyed by Chinese words and their English equivalents (Dong & Dong, 2006). Compared with WordNet, HowNet's architecture provides richer information apart from hyponymy relations. It also enriches relational links between words via encoded feature relations. The advantages of HowNet are (a) inherent properties of concepts are derived from encoded feature relations in addition to hypernymous concepts, and (b) information regarding conceptual differences between different concepts and information regarding morpho-semantic structure are encoded. HowNet's advantages make it an effective electronic dictionary for NLP applications.

The development of E-Hownet started in 2003. The set of primitives and taxonomy of HowNet and is adjusted to suit the goal of semantic composition. The current E-HowNet ontology is the result of automatic constructed by a computer program according to the pre-defined hierarchical structure of primitives and basic concepts as well as E-HowNet expressions for all words entries. E-HowNet extends a large set of basic concepts which make a deeper hierarchical structure and more precise semantic branching. It also results that lexical senses expressed based on basic concepts became more precise and readable. We also adjust the ontology structure into two parts. The first part is hierarchy for entities and the second part is hierarchy for relations, i.e. semantic roles. Furthermore the Attribute types and Value types are correspondingly organized.

Each word sense is a node of the taxonomy and expressed by an E-HowNet expression. Synonyms or near synonyms should be expressed by the same expression. Therefore E-HowNet ontology is formed by all lexical senses as well as primitive and basic concepts in a hierarchical order. Approximately 2,600 primitives from HowNet to form the top-level ontology of E-HowNet, which includes two types of subtrees: entities and relations. Entities indicate concepts that have substantial content. By contrast, relations play the role of linking semantic relations between entities. Any concept inherits all the fundamental features of its hypernym and must have at least one feature that its hypernym does not own.

Fig. 2 shows lexical example of "牛肉麵(beefnoodles)" and "炒飯(fried rice)" on Ehownet. Synonyms or near synonyms of "牛肉麵(beefnoodles)" include "烏龍麵(udon noodles)", "涼麵 (cold noodles)", 素麵(vegetarian noodle), etc. The lexical senses of these words are expressed based on basic concept – "麵 |noodles". And Synonyms or near synonyms of "炒飯" consist of "粥(gruel)", "竹筒飯(bamboo rice)", "咖哩飯(rice with curry sauce)", etc. The lexical senses of these words are expressed based on basic concept – "飯|CookedRice".

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

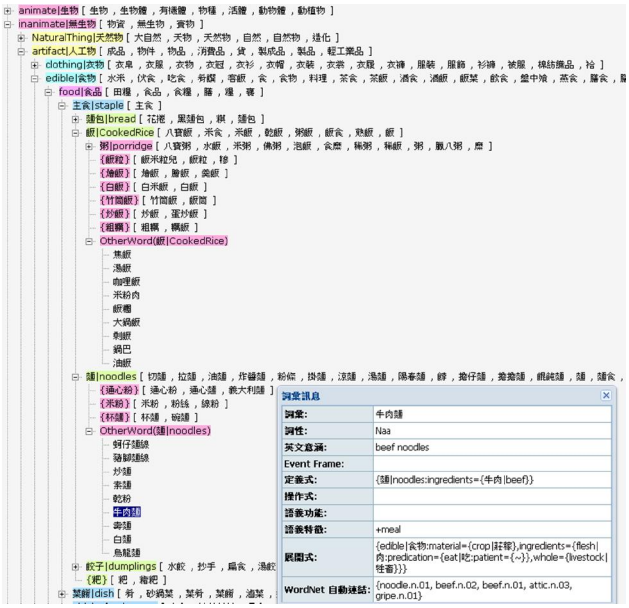

Figure 2. Lexical example of "牛肉麵(beefnoodles)" and "炒飯(fried rice)" on Ehownet.

3. Model

3.1 Concept Prediction

To predict the concept for each word of the given word sequence, we first need to predict the sense for each word in ENowNet, as known as word sense disambiguation (WSD). The challenge is there is no annotated corpus using sense definition of EHowNet available, so we can not obtain a sense disambiguate through a direct framework of supervised learning. To address this issue, we utilize the comprehensive part of speech (POS) defined in EHowNet and a Chinese corpus with annotations of simplified POS to achieve the effect of WSD. The approach is based on that a word's POS is highly related to its sense. The statement is based on our two observations: one is for almost all Chinese words, once a word's simplified POS is identified, its comprehensive POS can be referred into a comprehensive POS. The other observation is that for most cases in Ehownet, a pair of word and its comprehensive POS represents a unique sense. For example, Table 1 shows all POSs for noun and "牛肉麵(beefnoodles)"'s simplified and comprehensive POSs are "Na" and "Naa" respectively. While "牛肉麵_Na" is identified, "牛肉麵_Naa" can be inferred directly and this pair of word and POS only has a unique sense in EHowNet. The complete POS in EHowNet is listed in Appendix.

Simplified POS	Comprehensive POS	meaning
Na	Naa, Nab, Nac, Nad, Naea, Naeb	Normal noun
Nb	Nba, Nbc	Proper noun
Nc	Nca, Ncb, Ncc, Nce	Locational noun
Ncd	Ncda, Ncdb	Positional noun
Nd	Ndaa, Ndab, Ndc, Ndd	Time noun

Table 1 Simplified and Comprehensive POS for nouns in EHowNet

Through this strategy, WSD problem can be solved by a POS tagging problem for most cases, and we are able to use supervised training technique to solve POS tagging problem. For this task, we use Hidden Markov Model to carry on POS tagging.

To obtain concept sequence from sense sequence, we need to define which concept representation is suitable for this task and appropriate to address the data sparseness problem mentioned in Section 1. Since in EHowNet, all lexical senses are expressed based on primitive or basic concept, such as "牛肉麵(beefnoodles)", whose lexical sense is expressed as "{麵|noodles:ingredients={牛肉|beef}}", shown in Fig 2, it is natural to use primitive or basic concept as the concept representation for our goal. In the example, the concept of "牛肉麵(beefnoodles)" is simply "麵|noodles" or "noodles" for short.

Use the example to illustrate the whole process of concept prediction as follows.

Input: 吃 牛肉麵 還是 炒飯?

After Sense Prediction: 吃_VC31 牛肉麵_Naa 還是_Caa 炒飯_ Nab?

After Concept Prediction: eat noodles or rice?

3.2 LSTM with Concept Sequence

Given input word sequence, we follow the procedure of Concept prediction described in the previous section to obtain concept sequence. Then we use LSTM-based encoder and decoder to understand the post with the form of concept sequence and generate the response with the form of word sequence, as shown as Figure 3.

4. Experiment

Based on our training set, we experimented on four factors on LSTM encoder-decoder framework, including different seq-to-seq types, word embedding pretraining ways, attention models, and N-gram types. And the four combinations listed on Table 2 turned out to provide the best performance.

	LSTM Seq-to-Seq Type	Pretrain word embedding	Attention model type	N-gram on decoding
Run-G1	WS-to-WS	CBOW	general	bigram
Run-G2	WS-to-WS	no	general	bigram
Run-G3	WS-to-WS	CBOW	concat	trigram
Run-G4	CS-to-WS	CBOW	concat	bigram

Table 2. Four summited runs from CKIP.

On Table 2, "WS-to-WS" indicates "WordSequence-to-WordSequence" while "CS-to-WS" represents "ConceptSequence-to-WordSequence". Attention model type refers to the mapping function used in the attention model, including general and concat type defined in [3]. Word embeddings have 300 dimensions, and G-Run1, G-Run 2 and G-Run 4 use pretrained word embedding by using CBOW of word2vec on ASBC Chinese corpus with size of 10 million words.

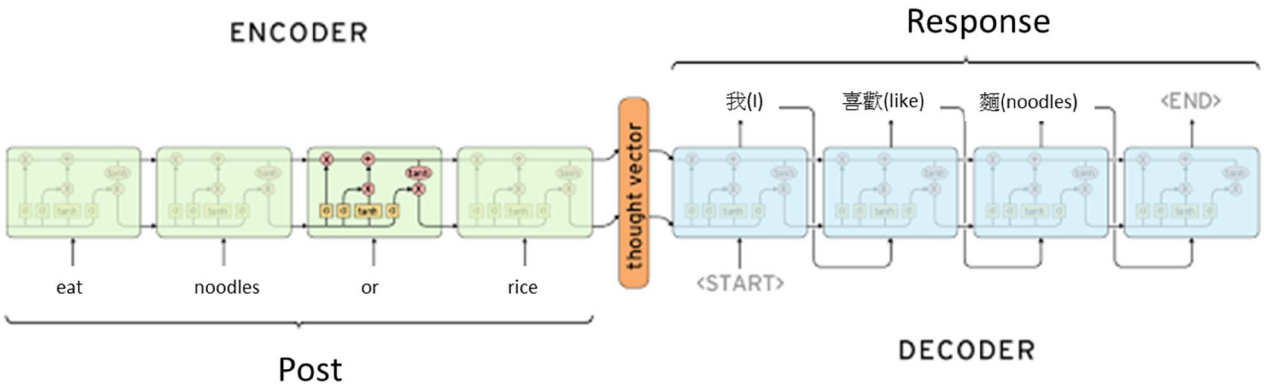

Figure 3. An LSTM model with concept sequence- "eat noodles or rice)" as its input and response"我喜歡麵(I like noodles)"> as its output. Diagram is modified from the original one at Google Research Blog.

The training dataset provided by task organization includes around 4,433,949 post-response pairs. Based on the introduction of the training set, 11,535 out of 4,433,949 post-response pairs have been labelled its quality by human annotators through the following procedure.

IF (fluent AND coherent)
  IF (self-sufficient AND substantial)
    assign L2
  ELSE
    assign L1
ELSE
  assign L0.

Based on the procedure, we can infer that the pairs with L2 have highest quality while the pairs with L0 have lowest quality. During training, we filter out the pairs which are labelled as L0 by all three annotators, leaving only 6276 pairs are used for training.

4.1 Result

From the announced results by task organization [11], all of our four runs do not get into top 90 out of all runs in this competition. Based on our observation, most of our generated responses still lack fluency and coherence. We conjecture the major reason lies on 6276 pairs are not sufficient to learn the language usage even though they have high quality. It seems that the whole 4,433,949 post-response pairs should be also used in some way in the training phase.

Although our preliminary experimental results are not satisfying in this task, we still attempt to show some interesting findings through the comparison among the four runs, as shown as Table 4.

	Mean MSnDCG@0001	Mean P-plus	Mean nERR@0010
Run-G1	0.0017	0.0029	0.0015
Run-G2	0.005	0.0086	0.0046
Run-G3	0.01	0.0171	0.0093
Run-G4	0.0083	0.0143	0.0077

Table 3. Comparison among four summited runs from CKIP.

From Table 3, it is obvious that Run-G3 and Run-G4 significantly outperform Run-G1 and Run-G2. Run-G3 slightly outperform Run-G4. However, we can not conclude that it is due to a single factor since their seq-to-seg type and N-gram on decoder are both different. But at least it is safe to say that CS-to-WS can preserve a certain degree of quality compared with WS-to-WS. It is worth investigating their comparison in performance under the same other factors in the future.

Post	好喜欢小葡萄的画啊[太开心] 喜欢的赶紧来围观哦[围观]
Run-G1	#、继续个冬至了
Run-G2	没有上海、幸福。精彩
Run-G3	你也要吃了吗？他不要过去的吗？
Run-G4	长的人心疼的图片好漂亮的好漂亮爆了。

Table 4. Case study of the test post – "test-post-10220"

Table 4 shows a case that Run-G4 outperforms Run-G3 in coherence but is worse than Run-G3 in fluency, which is concluded by that Run-G4 is more related to "画(painting)", but its decription - "好漂亮(very beautiful)" appears continuously, leading to poor fluency.

5. CONCLUSION

In this paper, we describe our main approach in this task along with a new idea of taking concept sequence as the input of the encoder for a LSTM-based encoder-decoder model in order to address the data sparseness problem in the training phase. The concept list is collected from EHowNet. With concept prediction, we obtain the concept for each word of the word sequence and thus form a sense sequence as the input of the LSTM model. Although our results are not among top systems in this task, the experimental results still show the potential of this idea through the comparison among our submitted runs. In the future, we will further investigate the effect of this idea under other STC conditions and improve the system performance by using larger number of post-response pairs as the training data.

6. REFERENCES

[1] Ryan Lowe, Nissan Pow, Iulian V. Serban and Joelle Pineau, The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructure Multi-Turn Dialogue Systems. SIGDial 2015.

[2] Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven Response Generation in Social Media. In EMNLP, pages 583–593.

[3] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-text Conversation. In Proceedings of ACL 2015. 1577-1586.

[4] Alex Graves. 2013. Generating Sequences with Recurrent Neural Networks. Preprint arXiv:1308.0850.

[5] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to Sequence Learning with Neural Networks. In NIPS, pages 3104–3112.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473

[7] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long Short-term Memory. Neural computation, 9(8):1735–1780.

[8] Dong Zendong, and Qiang Dong. 2006, HowNet and the Computation of Meaning. World Scientific Publishing Co. Pte. Ltd.

[9] Chen, Keh-Jiann, Shu-Ling Huang, Yueh-Yin Shih, and Yi-Jun Chen, 2005, Extended-HowNet- A Representational Framework for Concepts, OntoLex 2005- Ontologies and Lexical Resources IJCNLP-05 Workshop, Jeju Island, South Korea

[10] Huang, Shu-Ling, You-Shan Chung and Keh-Jiann Chen. 2008. E-HowNet: The Expansion of HowNet, the First National HowNet Workshop. Beijing, China.

[11] Lifeng Shang and Tetsuya Sakai and Hang Li and Ryuichiro Higashinaka and Yusuke Miyao and Yuki Arase and Masako Nomoto. 2017. Overview of the NTCIR-13 Short Text Conversation Task. Proceedings of NTCIR-13

Appendix

Part of Speech list in EHowNet

Simplified POS in EHowNet	Comprehensive POS in EHowNet	
A	A	/*非謂形容詞*/
Caa	Caa	/*對等連接詞，如：和、跟*/
Cab	Cab	/*連接詞，如：等等*/
Cba	Cbab	/*連接詞，如：的話*/
Cbb	Cbaa, Cbba, Cbbb, Cbca, Cbcb	/*關聯連接詞*/
Da	*Daa*	/*數量副詞*/
Dfa	Dfa	/*動詞前程度副詞*/
Dfb	Dfb	/*動詞後程度副詞*/
Di	Di	/*時態標記*/
Dk	Dk	/*句副詞*/
D	*Dab*, Dbaa, Dbab, Dbb, Dbc, Dc, Dd, Dg, Dh, Dj	/*副詞*/
Na	Naa, Nab, Nac, Nad, Naea, Naeb	/*普通名詞*/
Nb	Nba, Nbc	/*專有名稱*/
Nc	Nca, Ncb, Ncc, Nce	/*地方詞*/
Ncd	Ncda, Ncdb	/*位置詞*/
Nd	Ndaa, Ndab, Ndc, Ndd	/*時間詞*/
Neu	*Neu*	/*數詞定詞*/.
Nes	*Nes*	/*特指定詞*/
Nep	*Nep*	/*指代定詞*/
Neqa	*Neqa*	/*數量定詞*/
Neqb	*Neqb*	/*後置數量定詞*/
Nf	Nfa, Nfb, Nfc, Nfd, Nfe, Nfg, Nfh, Nfi	/*量詞*/

Ng	Ng	/*後置詞*/
Nh	Nhaa, Nhab, Nhac, Nhb, Nhc	/*代名詞*/
Nv	Nv1,Nv2,Nv3,Nv4	/*名物化動詞*/
I	I	/*感嘆詞*/
P	P*	/*介詞*/
T	Ta, Tb, Tc, Td	/*語助詞*/
VA	VA11,12,13,VA3,VA4	/*動作不及物動詞*/
VAC	VA2	/*動作使動動詞*/
VB	VB11,12,VB2	/*動作類及物動詞*/
VC	VC2, VC31,32,33	/*動作及物動詞*/
VCL	VC1	/*動作接地方賓語動詞*/
VD	VD1, VD2	/*雙賓動詞*/
VE	VE11, VE12, VE2	/*動作句賓動詞*/
VF	VF1, VF2	/*動作謂賓動詞*/
VG	VG1, VG2	/*分類動詞*/
VH	VH11,12,13,14,15,17,VH21	/*狀態不及物動詞*/
VHC	VH16, VH22	/*狀態使動動詞/
VI	VI1,2,3	/*狀態類及物動詞*/
VJ	VJ1,2,3	/*狀態及物動詞*/
VK	VK1,2	/*狀態句賓動詞*/
VL	VL1,2,3,4	/*狀態謂賓動詞*/
V_2	V_2	/*有*/
DE	/*的, 之, 得, 地*/	
SHI	/*是*/	
FW	/*外文標記*/	
COLONCATEGORY		/* 冒號 */