International Journal of

Computational Linguistics & Chinese Language Processing

中文計算語言學期刊

A Publication of the Association for Computational Linguistics and Chinese Language Processing

This journal is included in THCI Core, Linguistics Abstracts, and ACL Anthology.

易繫辭曰上古結繩而治後世聖人易之以書契百官以治萬民以察說文敘曰蓋文字者經藝之本宣教明化之始前人所以垂後後人所以識古故曰本立而道生知天下之至賾而不可亂也教化既萌文心雕龍則謂人之立言因字而生句積句而成章積章而成篇篇之彪炳

Vol.15 No.1 March 2010 ISSN: 1027-376X

International Journal of Computational Linguistics & Chinese Language Processing

International Journal of

Computational Linguistics & Chinese Language Processing

The editing of this journal is subsidized by Center for Humanities Research National Science Council in 2010.

Aims and Scope

International Journal of Computational Linguistics and Chinese Language Processing (IJCLCLP) is an international journal published by the Association for Computational Linguistics and Chinese Language Processing (ACLCLP). This journal was founded in August 1996 and is published four issues per year since 2005. This journal covers all aspects related to computational linguistics and speech/text processing of all natural languages. Possible topics for manuscript submitted to the journal include, but are not limited to:

- Computational Linguistics
- Natural Language Processing
- Machine Translation
- Language Generation
- Language Learning
- Speech Analysis/Synthesis
- Speech Recognition/Understanding
- Spoken Dialog Systems
- Information Retrieval and Extraction
- Web Information Extraction/Mining
- Corpus Linguistics
- Multilingual/Cross-lingual Language Processing

Membership & Subscriptions

If you are interested in joining ACLCLP, please see appendix for further information.

Copyright

Cover

Calligraphy by Professor Ching-Chun Hsieh, founding president of ACLCLP
Text excerpted and compiled from ancient Chinese classics, dating back to 700 B.C.
This calligraphy honors the interaction and influence between text and language

International Journal of
Computational Linguistics and Chinese Language Processing
vol. 15, no. 1, March 2010

Contents

Papers

The Association for Computational Linguistics and Chinese Language Processing

Computational Linguistics and Chinese Language Processing
Vol. 15, No. 1, March 2010, pp. 1-18

Harmony Graph, a Social-Network-Like Structure, and Its Applications to Music Corpus Visualization, Distinguishing and Music Generation

Wei-An Chen[*], Jihg-Hong Lin[+], and Shyh-Kang Jeng[#]

Abstract

In this research project, we propose a model, the Harmony Graph, to decompose music into a social-network-like structure according to its harmonies. The whole Harmony Graph network represents the harmony progressions in music. The Harmony Graph is utilized to visualize, distinguish, and generate music for four prepared corpora using social network techniques. We experimented on different characteristics in social network analysis, and we found significant differences among the Harmony Graphs of the four corpora. A new measure called *Agglomeration* is created to characterize the agglomerating phenomenon that cannot be described sufficiently by existing measures. A corpus-based music composition method is also proposed in this research. By performing random-walk in a Harmony Graph, we generated new music that differs from yet reflects the style of music pieces in the corpus. With the link prediction technique, we also generated music more pleasant aurally than simply using random walks.

Keywords: Social Network Analysis, Corpus Visualization, Corpus-Based Generation

[*] Computer Music Laboratory (JCMG), Graduate Institute of Communication Engineering, National Taiwan University
E-mail: r98942051@ntu.edu.tw

[+] Business & Marketing Strategy Labortory, Chunghwa Telecommunication Laboratories
E-mail: johnlin@cht.com.tw

[#] Department of Electrical Engineering and Graduate Institute of Communication Engineering, National Taiwan University
E-mail: skjeng@cc.ee.ntu.edu.tw

1. Introduction

Corpus is the basis of Corpus Linguistics. It is also important in Computer Music (Manaris *et al.*, 2007). In the research of corpus-based generation, such as generating articles from text corpora (Stribling *et al.*, 2009; Marom & Zukerman, 2005), speech synthesis based on audio corpora (Iida *et al.*, 2003), and generating music from music corpora (Conklin, 2003; Polashek *et al.*, 2005), one interesting topic is that the selection of corpora may lead to results with different styles. For example, an article generated from a corpus of Abraham Lincoln may reveal his style, which reads differently from one generated from a corpus of William Shakespeare. In this paper, we develop a new model, the Harmony Graph, to make music using corpus-based generation and use this model for music-corpus distinction.

The Harmony Graph is applied to organize a music corpus into a social-network-like structure, analogous to the Word Graph (Oerder & Ney, 1993) in Corpus Linguistics. Four distinct music corpora were prepared that are collections of music in different genres.

In Corpus Linguistics, there has been relevant research on text corpus visualization (Paley, 2002; Fortuna *et al.*, 2005; Rohrer *et al.*, 1998), which generally provides the overall concept of the corpora. Nevertheless, they cannot be used to tell corpora apart at a glance. In the area of Computer Music, distinction of music pieces into different genres using Harmony Graphs is found to be accurate and believed to be new.

In addition to visual inspection, we apply social network analysis to these Harmony Graphs. The calculated measures, namely degree distribution, average path length (APL), and clustering coefficient (CC), indicate that social network analysis is very useful for distinguishing corpora. We also devise a measure, Agglomeration, to capture the density of connections in the graph. This new measure is found to be even more helpful in distinguishing music corpora.

In corpus-based music generation, we begin by performing a random-walk in a Harmony Graph to generate music. For zero-occurrence smoothing, we apply the link prediction technique in social network methodology to add potential edges, and increase the variety of produced music. The generated music somewhat reflects the style of selected corpus according to results of subject tests. Although some relevant research regarding regeneration of music styles has been published (Dubnov *et al.*, 2003; Trivino-Rodriguez & Morales-Bueno, 2001; Pachet, 2003), the Harmony Graph model stands out for being visualizable, analyzable and interpolable by social network methodology.

Sequential music pattern mining has also been applied to model music by finding important sequences sampled from a database and using the model to classify or generate music (Shan *et al.*, 2002). Harmony Graph, however, is an approach quite different from pattern mining. After constructing the Harmony Graph, the music content is reduced from sequential data into a folded form, and attributes like APL, CC, and Agglomeration can be retrieved from it, based on social network analysis techniques. These attributes cannot be retrieved from pattern mining models directly.

To quantitatively evaluate the Harmony Graph's distinguishing ability, we built up a classifier according to the results of the social network analysis. Given a music input, the classifier predicts which corpus it belongs to by the social network features of its Harmony Graph. The accuracy of our experimental result is 73% out of 59 songs. As for the evaluation of corpus-based music generation, we conducted a subject test. In about 70% of the test queries, subjects agree that the generated music matches its corresponding corpus best in style among five choices. A demonstration program is also provided on the Internet for free download.

2. Experimental Setting and Model

2.1 The Four Corpora

In this study, we prepared four corpora, namely polyphony, homophony, pantonality, and atonality. Polyphony and homophony are tonal music, and the other two belong to atonal music. These corpora will be used for visualization, social network analysis, and corpus-based music generation. Table 1 shows the details.

Table 1. Four corpora used in this paper

Genre	**Composer** and Works
polyphony	**Bach** Inventions, Sinfonias, preludes, fuga
homophony	**Chopin** etude, ballade **Mendelssohn** Songs Without Words
pantonality	**Prokofiev** toccata, prelude, sonata **Shostakovitch** toccata, prelude, sonata
atonality	**Schoenberg** Klavierstücke

2.2 The Harmony Graph

Western music evolved from modal music in the Middle Ages to polyphonic music, glorified by Bach at its peak, and gradually became homophonic, which is music with melody accompanied by chords. The development of Harmony has been mature. As time progressed into the 20th century, the breakdown of tonality led to escape from harmony rules. In this research, we do not refer "harmony" as "chords" in classical Harmony. Rather, in a wider sense, we refer harmony as "the notes played simultaneously."

In this sense, we build a graph from music accordingly, which is named Harmony Graph. A node of a Harmony Graph is a harmony represented by a string of note names, *e.g.* "D F #A". The octave information is suppressed, which means that both C1 and C2 are regarded as the same, and are notated as C. The links of a Harmony Graph represent note changes, that is, the progression of harmonies. Notice that, for simplicity of explanation, Harmony Graph here does not contain any temporal information, such as beat and rhythm, but only the progression of harmonies, namely pairs of harmonies that are temporally neighbored.

In addition, we create a "null" node, at which the music starts and ends. The music starts from null to the first harmony, and ends from the last harmony to null. Null also represents rests, where no notes occur.

Harmonies are encoded as a 12-bit binary number, corresponding to the twelve tones in an octave. For example, "000000000001" represents C, "001000000010" represents "A C#", and so on. There are 4096 possible combinations of all harmonies. Hence, each harmony can also be represented by an integer from 0 to 4095, including the null node “000000000000”.

The weight of each link represents the number of times that the same progression happens. For example, the more harmony A to harmony B occurs in a piece of music, the higher the weight of the link AB will be.

We use MIDI as raw data format. To simplify the problem, we consider only the onset time, offset time, and the pitch position of each MIDI event.

Figure 1 is an example of how to build a simple Harmony Graph. Three steps are required to construct the Harmony Graph of one music piece:

Step 1. Extracting notes.

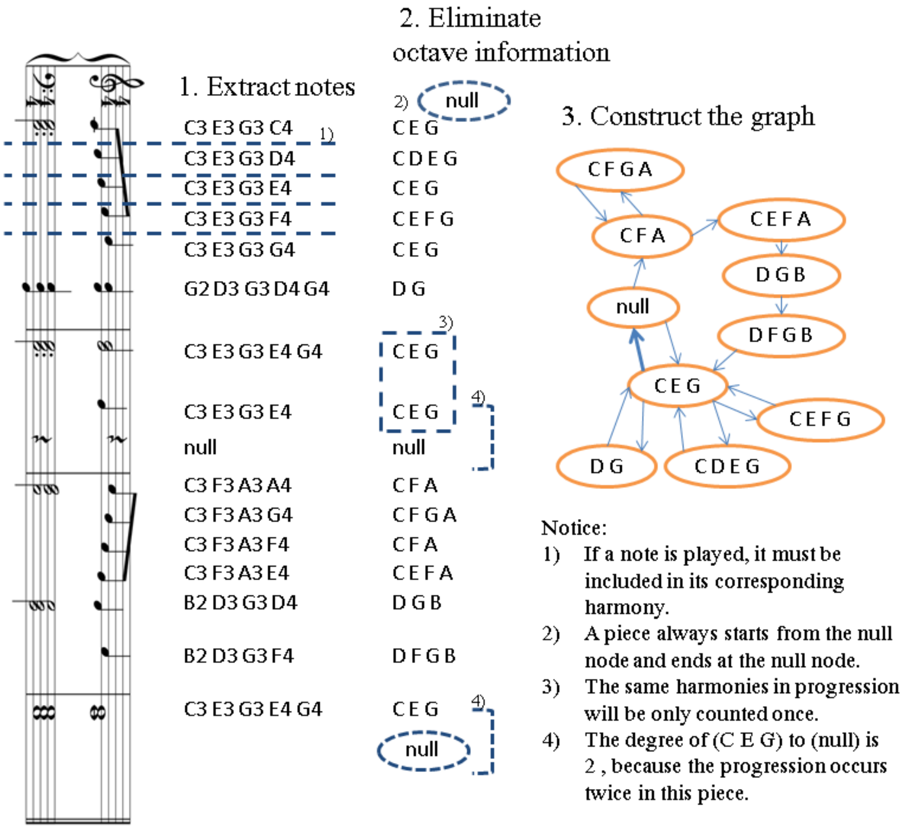
Scan the sheet music along the time line and record notes happening at the same time as a harmony. As soon as a note combination changes, a new harmony is generated and recorded.

Step 2. Suppressing octave information.

Suppress the octave information of the harmonies obtained in Step 1, and merge the notes with the same note name. For example, the first harmony (C3 E3 G3 C4) becomes (C E G), because C3 and C4 are both C, just in different octaves.

Step 3. Constructing the graph.

Connect the harmonies in Step 2 according to their sequential order. Then, link the null node to the first harmony, and link the last harmony back to the null node. Furthermore, rest notes in the music piece are treated as the null node. After connecting the harmonies and the null node, a Harmony Graph is accomplished.



Notice:
1) If a note is played, it must be included in its corresponding harmony.
2) A piece always starts from the null node and ends at the null node.
3) The same harmonies in progression will be only counted once.
4) The degree of (C E G) to (null) is 2 , because the progression occurs twice in this piece.

Figure 1. A simple example of constructing a Harmony Graph

3. Results

3.1 Corpus Visualization

Graphviz (Ellson *et al*., 2002) is applied to visualize a Harmony Graph. We have found that its built-in fdp engine is especially suitable for drawing graphs, because the higher-degree nodes will be placed closer to the center and the lower-degree nodes closer to the boundary. This makes it easier to observe the characteristics of the graph.

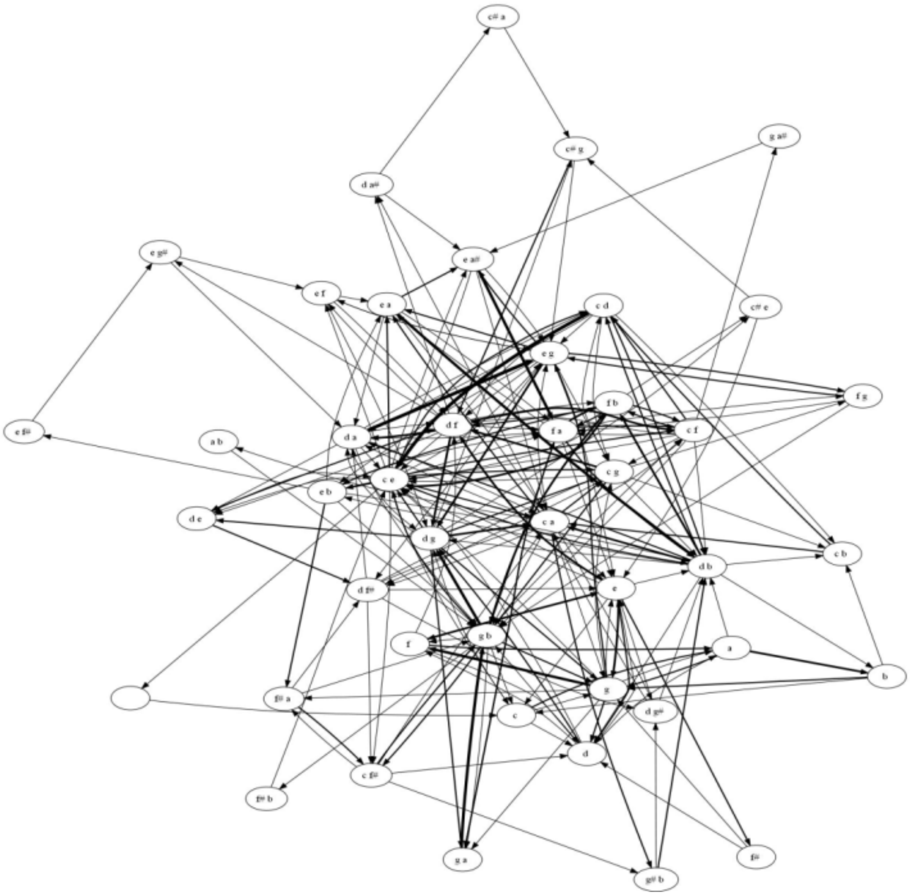


Figure 2. Visualized Harmony Graph of Bach's Invention No. 1, a polyphonic piece

Figures 2 to 5 show some representative outcomes of our corpora from four genre of music.

Figure 2 is derived from Bach's "Invention" No. 1, and is a representative Harmony Graph for polyphonic music. We find:

The number of notes in each single node is at most two, because "Invention" is two-part polyphonic music, like a dialogue between two melody lines. Therefore, there are a maximum of two notes at the same time. The upper bound of node number is 79 for two-part polyphonic music, since

$$C_2^{12} + C_1^{12} + 1 = 79. \tag{1}$$

Near the center of the picture, nodes are connected with each other in a very complicated way. We call this agglomeration, which will be discussed further in Section 3.2.4.

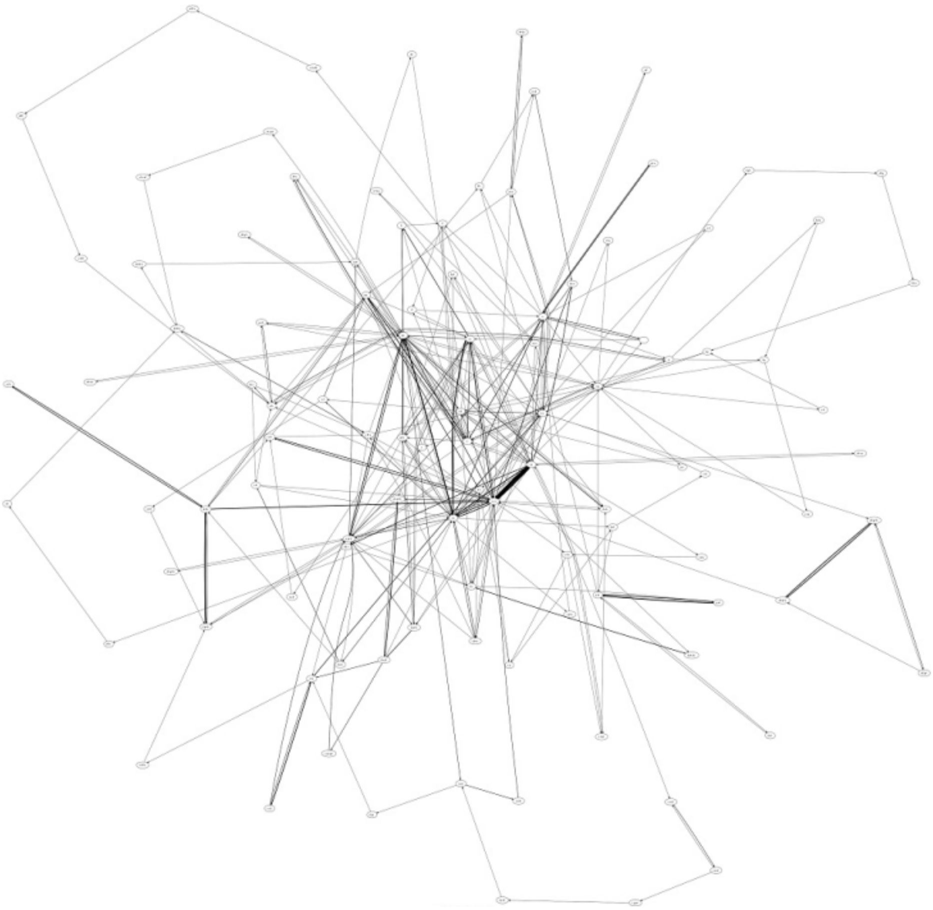


Figure 3. Visualized Harmony Graph of Mendelssohn's Song Without Words, Op. 19-2, a homophonic piece

Close to the border of the picture, a small number of nodes have only one incoming link and one outgoing link. This means that these harmonies are used only once in the whole masterpiece. These harmonies tend to be special ones used by the composer.

Figure 3 is the Harmony Graph of "Song Without Words", Op. 19-2 by Mendelssohn, and is a representative for homophonic music. It has a larger scale with more nodes than Fig. 2 has. And the phenomenon of agglomeration is also obvious. In the periphery, however, there are more lower-degree nodes. This may stand for the more freedom of harmony usage, compared with polyphonic music. Furthermore, unlike in Figure 2, we can find a thick link in Fig. 3. Similar links are also found in other graphs. We infer that this thick link is the outbound of the tonal center.

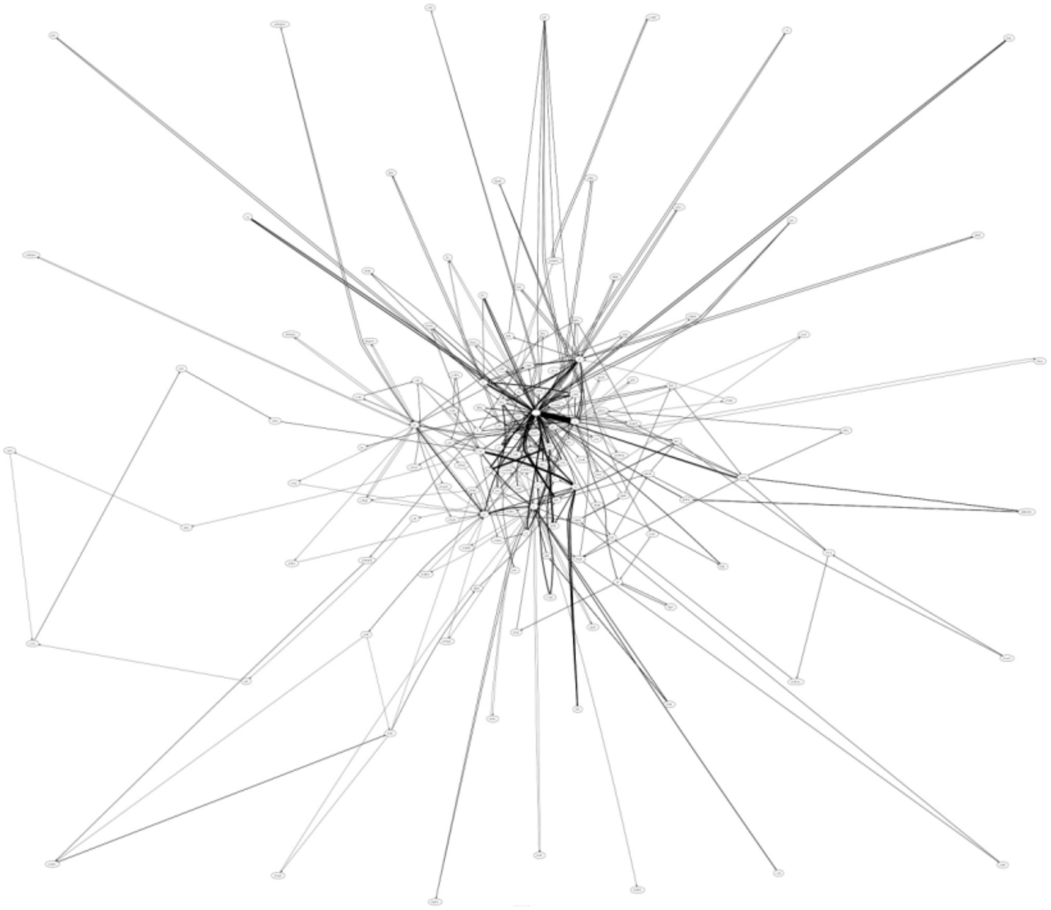


Figure 4. Visualized Harmony Graph of Prokofiev's Sonata, Op. 14, Movement 2, a pantonal piece

Figure 4 is the picture of Sonata, Op. 14, Movement 2, by Prokofiev, which is a pantonal piece. Apparently, the number of peripheral nodes is much more than in the previous two figures. This is an indication of atonal music and the increased freedom in usage of harmonies. The agglomeration is less obvious, which means that the treatment of harmonies is less confined than traditional music. We also can see thick links near the center. After inspection of Prokofiev's graphs in general, we find that the thick links are mostly linked to the null node, which may be an indication that Prokofiev treats the piano as a percussive instrument.

Figure 5 shows Schoenberg's "Klavierstücke", Op. 19-5, which is atonal piece. Compared with the previous ones, there is almost no agglomeration, which means a more distant relationship among harmonies.

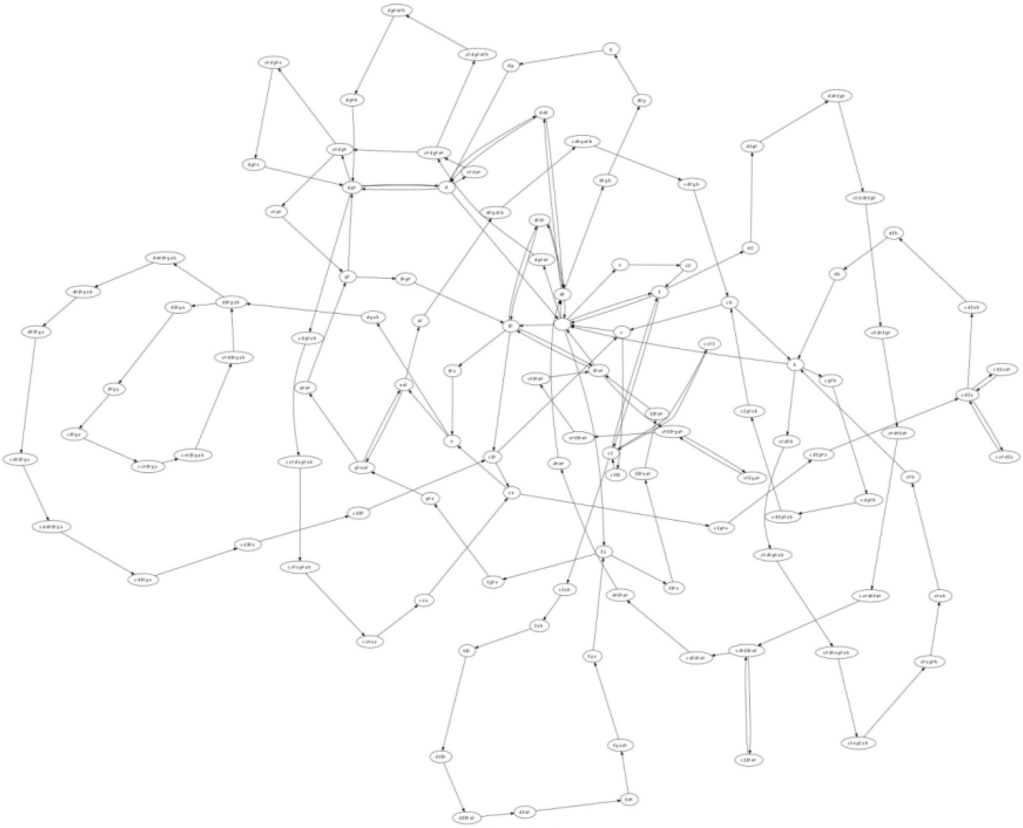


Figure 5. Visualized Harmony Graph of Schoenberg's Klavierstücke, Op. 19-5, an atonal piece

These phenomena reveal that Schoenberg's composition method deviates completely from the norms of traditional Harmony.

3.2 Social Network Analysis

In this section, we apply social network analysis techniques to examine the Harmony Graphs of the four corpora. Their degree distribution, average path length, and clustering coefficient are discussed in the following three subsections, respectively. Then, we introduce a newly proposed measure, *Agglomeration*, to describe the agglomeration phenomenon.

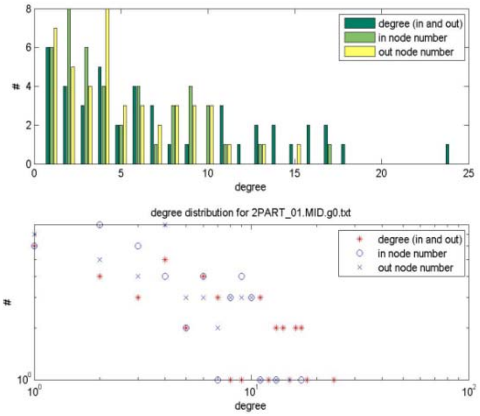


Figure 6. Degree distribution of two-part polyphonic piece

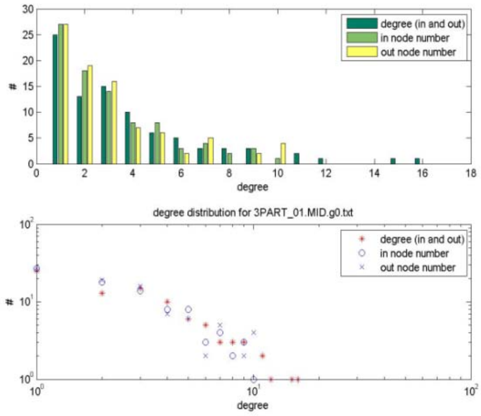


Figure 7. Degree distribution of three-part polyphonic piece

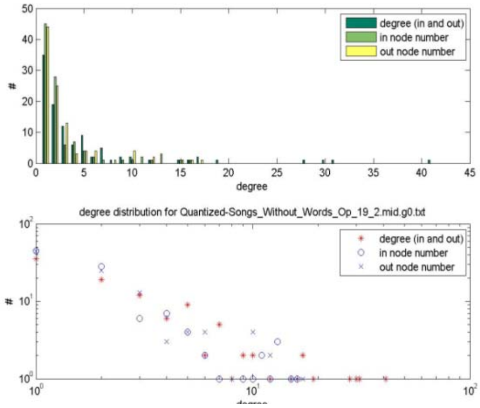


Figure 8. Degree distribution of homophonic piece

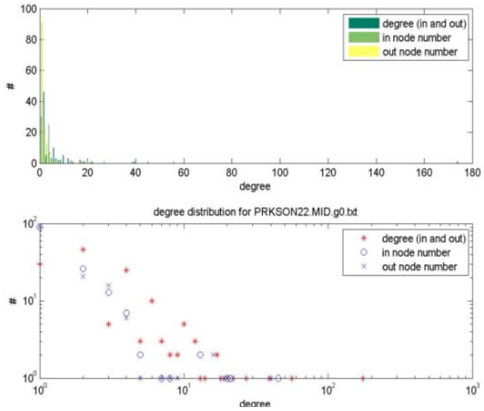


Figure 9. Degree distribution of pantonal piece

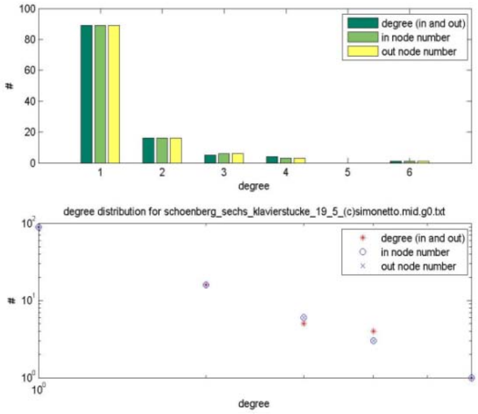


Figure 10. Degree distribution of atonal piece

3.2.1 Degree Distribution

Figure 6 to Figure 9 are the degree distributions of the four Harmony Graphs corresponding to Fig. 2 to Fig. 5, respectively. In each figure, the upper bar chart shows the degree distribution of weighted degree, unweighted incoming degree, and unweighted outgoing degrees in linear scale. As previously mentioned, the weighting is the count of the occurrences of edges. The lower scattered chart shows the same data in logarithmic scale to examine if it fits the Power Law. From those results, we see that all of them follow the Power Law except Fig. 6, Bach's two-part “Inventions”. At first glance, the reason might be that there are not enough nodes, since the Harmony Graphs of the three-part “Sinfonia”, which all follow the Power Law, have more nodes. Nevertheless, there are also very few nodes in Fig. 10, which still meets the Power Law. The same result applies in all of the other masterpieces of this genre.

Therefore, we speculate that the Harmony Graph follows the Power Law in normal circumstances, but for two-part polyphonic music such as “Invention”, the Power-Law effect is weaker due to strong tonality and node scarcity. This conjecture requires further in-depth investigation.

3.2.2 Average Path Length

In Section 3.1 we mentioned that there exist "long bridges" in the Harmony Graphs of 12-tone serial works. This can be best described in terms of the average path length (APL). Actually, in our experiments, we find that APL is the most significant characteristic to distinguish musical styles.

For two-part polyphonic music, APLs are normally under 3 due to fewer nodes and a high degree of agglomeration. As the music becomes more complex, for three-part polyphony and homophonic music, APL is slightly larger, between about 3 to 4. For non-tonal music, which was composed by numerous and various techniques, the corresponding APL has the most deviation, varying from 2 to 7. For Twelve-tone series works, all of the APLs are larger than 6.

3.2.3 Clustering Coefficient

In traditional social network analysis, the Clustering Coefficient (CC) is mostly relevant to characterize the aforementioned agglomeration phenomenon. For the most agglomerated Harmony Graph in our experiments, the two-part polyphonic music, the CC is about 0.3 to 0.4. For the other types of music, the CC is relatively smaller, about 0.001 to 0.1. Generally, CC alone is insufficient to distinguish the corpora, but when used in conjunction with other measures, the results are useful.

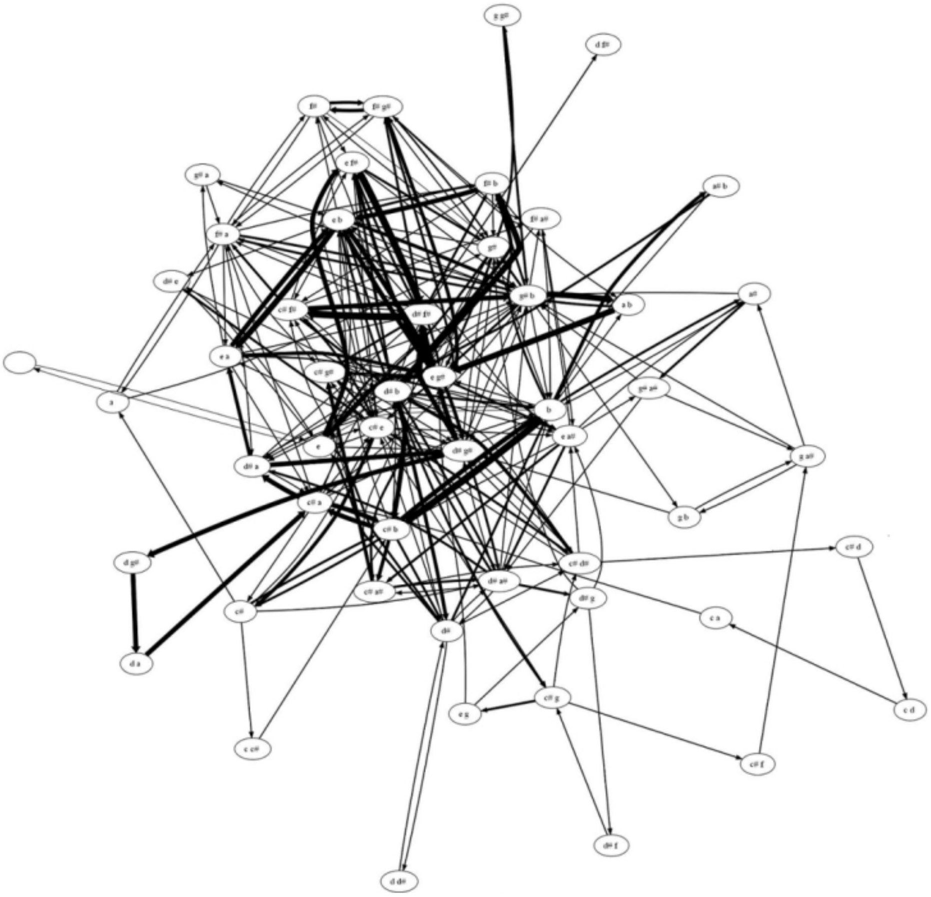


Figure 11. Visualization of Harmony Graph of Inventia 6.

In some cases, the CC does not confirm with agglomeration. For example, the Harmony Graph of “Invention” 6 in Fig. 11 has CC = 0.003, which is relatively small, but we can see that the nodes are strongly bonded with each other.

Since CC is calculated through the number of triangles, which is not necessarily related to bonding, a more reliable measure for explaining this phenomenon is needed.

3.2.4 Agglomeration

After studying Figure 2 to Figure 11, we found that the agglomeration phenomenon occurs when high-degree nodes link together, in contrast to the conditions for a large CC, which is due to large number of triangles formed by clusters of links in the graph. We thus propose an *Agglomeration* measure (*agg*):

$$agg = \frac{\sum_{i,j\in G} D(i)D(j)\delta_{ij}}{[\sum_{\mathrm{i}\in \mathrm{G}} D(i)]^2}, \qquad (2)$$

where δ_{ij} denotes the adjacent status between nodes i and j. If nodes i and j are adjacent to each other, $\delta_{ij} = 1$, otherwise $\delta_{ij} = 0$. Notation $D(x)$ represents the degree of node x. By design, if high-degree nodes connect with each other, the corresponding *agg* value will be large.

Equation (2) can also be rewritten as:

$$agg = \sum_{i,j \in G} \frac{D(i)}{m} \frac{D(j)}{m} \delta_{ij}, \tag{3}$$

where m is the total degree of the graph. From (3), we can see that *agg* is also the probability showing the likelihood that two randomly-chosen nodes are adjacent. We can verify that, if the high-degree nodes are linked with each other, the probability that an adjacent node pair is selected is higher. Note that the range of this measure is from 1 for a complete graph down to 0 for a completely isolated graph.

In our experiments, we find *agg* is more suitable than CC to describe agglomeration. For instance, "Invention" 6, an especially agglomerative case, has an *agg* of 0.25, which is noticeably higher than the average *agg* of all "Inventions". On the other hand, its CC is 0.003, which is far below the average CC of all "Inventions".

Generally speaking, *agg* represents the degree of relation between harmonies. The *agg* of the Harmony Graphs we studied varies from 0.05 to 0.4. For a genre with strong harmony relations such as tonal music, *agg* tends to be large, and *vice-versa*. Nevertheless, we should not take *agg* as a measure of the degree of tonality, because non-tonal music might also have some strong harmony relations, such as modal music.

3.3 Corpus-Based Music Generation

3.3.1 By Random Walk

In a Harmony Graph each node represents a harmony; therefore, one directed edge binds two harmonies, and can be treated as a harmony progression. If we walk randomly in the Harmony Graph, the resultant harmony progression can produce music. We call this Graph Music.

For music generation, the build-up of the Harmony Graph is slightly extended. We not only need to save the count of occurrences of the harmony progressions as the weighting of edges, but also the durations. Thus, each edge is additionally tagged with a duration, such as a quarter note or a sixteenth, according to the learned data. Then, during the random walk, the random walker can pick among edges of different durations. So, the duration of the random chord progression is also randomly picked, and the produced music is rhythmic.

One feature of Graph Music is that it can reproduce similar music styles. We constructed a demonstration program that is harnessed with different Harmony Graphs built from masterpieces of Bach, Mendelssohn, Chopin, Brahms, Prokofiev, Shostakovich, and Schoenberg. If we switch among different composers, we can hear the style of the generated Graph Music changing accordingly, because the Harmony Graph has the effect of shuffling the corpus evenly, while reserving the most important information about the styles. Thus, the produced music sounds novel yet familiar.

3.3.2 By Link Prediction

The preliminary version of Graph Music has a drawback. During the random walk, if the degree of the current node is 1, there is only one choice for the next node. It is very likely that the next node also has degree of 1 if the portion of the corresponding original music in the corpus is quite unique, thereby trapping the random walker. The longer the path with such nodes, the more the produced music sounds like just a copy of the original music. It is analogous to the zero-occurrence problem in Corpus Linguistics. Here, we utilize the "link prediction" technique in social network for improvement.

Link prediction estimates the probability of connection for two unconnected nodes. When our random walker departs from one node, we make it choose some other unconnected nodes as extra candidates according to their link prediction probability. The estimated probability that two harmonies are linked is derived from their similarity. We believe similar harmonies have better continuity.

For two harmonies, A and B, we define the similarity as the number of their common notes divided by the number of notes in each harmony:

$$\text{similarity}(\mathrm{A},\mathrm{B}) = \frac{|A \cap B|}{|A||B|}. \tag{4}$$

Note that the result will range between 0 and 1, inclusively. Then, we define the link prediction probability that an edge connecting from node S to node T exists in (5).

$$\text{prob}\big((\mathrm{S},\mathrm{T}) \in \mathrm{E}\big) = \max\{ \max_{(\mathrm{S},\mathrm{i})\in \mathrm{E}} \text{similarity}(\mathrm{i},\mathrm{T}), \max_{(\mathrm{j},\mathrm{T})\in \mathrm{E}} \text{similarity}(\mathrm{S},\mathrm{j})\}. \tag{5}$$

Here, E denotes the set of edges. By (5), we first find the outbound node of S with the highest similarity to T. We also find the inbound node of T with the highest similarity to S. Then, we pick the larger similarity value between the two as the link prediction probability. The logic is “Since S connects to a node similar to T, it is likely that S also connects to T.” or “Since a node similar to S connects to T, it is likely that S also connects to T”. Since we added soft links to harmonies with good continuity, the new music demonstrated more variety without abrupt changes.

4. Evaluation

4.1 Corpus Distinguishing

The qualitative discussion in Section 3.2 gives us some insight about different corpora. So, in this section, we use the four attributes discussed in Section 3.2 to perform supervised learning to verify how well we can differentiate between different corpora. The classifier we use here is SVM. The music entries and their corresponding categories are shown in Table 2. They were MIDI files mainly collected from the websites Classical MIDI Connection and kunstderfuge.com. Using the toolkit LIBSVM (Chang & Lin, 2001), with experimental settings cost equals 4, and gamma equals 1/70, the accuracy out of 59 entries in a 5-fold cross validation is 73%, which shows pretty good performance of this new model in classification.

Table 2. Five categories used in SVM test.

Genre	**Composer** and Works
2-part polyphony	**Bach** Inventions
3-part polyphony	**Bach** Sinfonias,
homophony	**Mendelssohn** Songs Without Words
pantonality	**Prokofiev** toccatas, preludes, sonatas
atonality	**Schoenberg** Klavierstücke

4.2 Corpus-Based Music Generation

The evaluation of the produced music to see if it follows a specific style is very subjective. Therefore, we provide a downloadable demonstration program for readers to rate it in person[1].

Note that users can also test on their own corpora by adding distinct folders of MIDI files. See the included instruction file for more details.

In addition, we conducted a subject test to show that the Graph Music somehow reflected the styles of the corpora. We set up a website to allow online testing and collected 245 responses from 21 participants. For each independent test, the participant would listen to a piece of music generated from one out of the five corpora of different composers, namely Bach, Mendelssohn, Brahms, Schoenberg, and Shostakovich. Then, original masterpieces of each composer were provided for comparison. The participants just listened to these six pieces of music, without any other information such as the name of the song or the composer. The

[1] URL for Graph Music program, http://homepage.ntu.edu.tw/~d96944001/GraphMusic

participant was then asked to choose one among the five original masterpieces such that the selected music is closest to the generated piece in style. After answering this question, the participant could decide to take one more independent test or just stop.

Our theory behind the experiment is as following. If the music was unrelated to the style, the participant could answer only by random guessing, hence, the accuracy should be about 20%. On the other hand, if the accuracy is greater than random guess, it indicates that there exists some recognizable relation behind the generated music and its corresponding corpora. To study the general case, we chose the participants from friends and classmates who have no advanced music background, *i.e.*, the participants were not familiar with those composers' works.

The collected responses are shown as the confusion matrix in Table 3.

Table 3. Confusion matrix of subject test.

	Answer					
Question	**Bach**	**Mendelssohn**	**Brahms**	**Schoenberg**	**Shostakovitch**	Accuracy
Bach	49	1	3	0	1	90.74%
Mendelssohn	10	36	2	2	2	69.23%
Brahms	4	3	42	2	0	82.35%
Schoenberg	1	1	0	44	2	91.67%
Shostakovitch	3	1	0	10	36	72.00%

In the matrix, the row represents the corpora the query music is generated from, and the column represents the answers from all participants. For example, the second row shows that among the 52 Mendelssohn questions, 10 were answered to be Bach, 36 were answered to be Mendelssohn (correct), and 2 for each of the other composers, which indicates an accuracy of 69.23%. So, we can assert that the generated music somehow reflects the styles of the corpora.

In statistical hypothesis testing, for all categories, the null hypothesis "the accuracy is 20% (due to random guessing)" was rejected and the alternative hypothesis "the accuracy is more than 20%" was accepted, with all confidence more than 99.9%, assuming that the accuracies were independent random variables following the student's *t*-distribution.

5. Conclusions

A social-network-like structure, Harmony Graph, for a music corpus, and with special emphasis on corpus distinction and music generation has been proposed. We prepared four music corpora of different genres, and derived Harmony Graphs for each corpus. The experiments show that the visualization of Harmony Graph is a good way to tell corpora apart. To be quantitative, we applied social network techniques to analyze Harmony Graphs. A new measure, *Agglomeration,* was also given to assess the strength of the relations between harmonies. To show the effect of corpus distinction in corpus-based music generation, we also provided a demo program for download. A subject test was also conducted in support of that the generated music somehow reflected the styles of the corpora.

Acknowledgments

We would like to thank Prof. Shou-De Lin for his detailed introduction to the research field in the Social Network Analysis course during 2009 fall. This research can not be done without Prof. Lin's advice. We also want to thank Prof. Shyh-Kang Jeng, director of NTU computer music lab. His subtle insight into computer music and valuable discussions has inspired us and besteaded a lot. In this research we utilized Stephen Toub's MIDI library for DotNet[2], which is an excellent tool for MIDI manipulation. Special thanks to him.

References

Conklin, D. (2003). Music generation from statistical models. In *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences,* 30-35.

Dubnov, S., Assayag, G., Lartillot, O. & Bejerano, G. (2003). Using machine-learning methods for musical style modeling. *Computer,* 73-80.

Ellson, J., Gansner, E., Koutsofios, L., North, S. & Woodhull, G. (2002). Graphviz open source graph drawing tools. In *Graph Drawing,* 594-597.

Fortuna, B., Grobelnik, M. & Mladenic, D. (2005). Visualization of text document corpus. *Special Issue: Hot Topics in European Agent Research I Guest Editors: Andrea Omicini*, *29*, 497-502.

Iida, A., Campbell, N., Higuchi, F. & Yasumura, M. (2003). A corpus-based speech synthesis system with emotion. *Speech Communication*, 40(1-2), 161-187.

Manaris, B., Roos, P., Machado, P., Krehbiel, D., Pellicoro, L. & Romero, J. (2007). A corpus-based hybrid approach to music analysis and composition. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, 22, 839.

[2] http://blogs.msdn.com/toub
E-mail: stoub@microsoft.com

Marom, Y. & Zukerman, I. (2005). Corpus-based generation of easy help-desk responses.

Oerder, M. & Ney, H. (1993). Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *icassp,* 119-122.

Pachet, F. (2003). The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3), 333-341.

Paley, W. (2002). TextArc: Showing word frequency and distribution in text. In *Poster presented at IEEE Symposium on Information Visualization*, Volume 2002.

Polashek, T., Miranda, E., Hay, J., Bauer, L. & Gunkel, D. (2005). Beyond Babble: A Text-Generation Method and Computer Program for Composing Text, Music and Poetry. *Leonardo Music Journal*, 15(1), 17-22.

Rohrer, R., Ebert, D. & Sibert, J. (1998). The shape of shakespeare: Visualizing text using implicit surfaces. In *Proceedings of the 1998 IEEE Symposium on Information Visualization,* 121-129.

Shan, M.-K., Kuo, F.-F. & Chen, M.-F. (2002). Music style mining and classification by melody. In *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, 1, 97-100.

Stribling, J., Krohn, M. & Aguayo, D. (2009). Scigen-an automatic cs paper generator. *Última visita*, 4, 12.

Trivino-Rodriguez, J. & Morales-Bueno, R. (2001). Using multiattribute prediction suffix graphs to predict and generate music. *Computer Music Journal*, 25(3), 62-79.

Computational Linguistics and Chinese Language Processing
Vol. 15, No. 1, March 2010, pp. 19-36

基於對照表以及語言模型之簡繁字體轉換

Chinese Characters Conversion System based on Lookup Table and Language Model

李民祥[*]、吳世弘[*]、曾議慶[*]、楊秉哲[+]、谷圳[+]

Min-Hsiang Li, Shih-Hung Wu, Yi-Ching Zeng,

Ping-che Yang, and Tsun Ku

摘要

中國大陸與台灣的文字同屬於華文字體，但字體上卻分為簡體字與繁體字。中國大陸與台灣近年來在中文書籍及網路上皆有大量的資訊交流。基於閱讀習慣，文字勢必需要執行簡繁轉換後才利於雙方的讀者閱讀。傳統的簡繁轉換擁有簡體一字對繁體多字的歧異問題以及兩岸用語不同的問題。因此，本研究設計一個具有擴展性的簡繁轉換系統，透過人工擷取維基百科新增對照表內容來改善兩岸用語不同的問題，以及使用語言模型改善簡體字一個字對繁體字多個字的歧異問題。此系統可以降低各種中文電子書籍執行簡繁轉換後人工校正的成本。具有彈性的架構使得系統可以持續擴充改進。

關鍵詞：簡繁轉換，語言模型，維基百科，對照表

Abstract

The character sets used in China and Taiwan are both Chinese, but they are divided into simplified and traditional Chinese characters. There are large amount of

[*]朝陽科技大學資訊工程系, Department of Computer Science and Information Engineering, Chaoyang University of Technology
E-mail: {s9827608, shwu, s9927605}@cyut.edu.tw
The author for corrrespondence is Shih-Hung Wu.
[+]資訊工業策進會, Institute for Information Industry
E-mail: {maciaclark,cujing}@iii.org.tw

information exchange between China and Taiwan through books and Internet. To provide readers a convenient reading environment, the character conversion between simplified and traditional Chinese is necessary. The conversion between simplified and traditional Chinese characters has two problems: one-to-many ambiguity and term usage problems. Since there are many traditional Chinese characters that have only one corresponding simplified character, when converting simplified Chinese into traditional Chinese, the system will face the one-to-many ambiguity. Also, there are many terms that have different usages between the two Chinese societies. This paper focus on designing an extensible conversion system, that can take the advantage of community knowledge by accumulating lookup tables through Wikipedia to tackle the term usage problem and can integrate language model to disambiguate the one-to-many ambiguity. The system can reduce the cost of proofreading of character conversion for books, e-books, or online publications. The extensible architecture makes it easy to improve the system with new training data.

Keywords: Chinese Character Conversion, Language Model, Wikipedia, Lookup Table.

1. 緒論

由於中國大陸與台灣數位出版合作的開啓，中文書籍相互流通的機會增加，簡體字與繁體字的轉換技術開始變得重要。近年來，隨著台灣與中國大陸兩岸交流逐漸頻繁，以及網路資訊快速發展，文字書信已經成爲兩岸溝通的媒介之一。然而，由於中國大陸普遍使用簡體中文，台灣主要使用繁體中文，因此雙方在文字溝通上勢必要先經過簡繁轉換的程序後才利於閱讀。一般來說，簡繁轉換是依據簡繁字對照表來進行轉換，這個方法主要使用單字詞一對一的方式進行簡繁轉換。不過在許多情況下，簡體字對應繁體字經常是一對多的狀況，所以僅使用一對一的方式進行轉換常常會出現字詞不適用的狀況，稱此爲「非對稱簡繁字」。

中國大陸以及台灣已著手研究簡繁轉換工具的有：中國大陸的中國科學院軟體所、四通利方資訊技術有限公司、新天地公司；台灣的 IBM 公司、倚天資訊股份公司，以及其它研發團隊等。目前也有許多文書處理軟體包含著內建的簡繁轉換系統，例如：Microsoft Office、Sun 的 OpenOffice；以及網路上可查詢到的雙語字典，例如：Google Translate。然而，這些轉換的結果通常參差不齊，簡繁轉換後依然需要依靠人工來校正不精確轉換的錯誤(王曉明、魏林梅，2008)。根據文獻，王寧擷取了 150 萬字的小說簡體字語料，使用 Office Word2003 執行簡體轉換繁體的功能，發現許多簡體字對繁體字一對多的情況無法正確被轉換(王寧、王曉明，2005)。簡而言之，簡繁轉換的困難在於簡體字存在著非對稱簡繁字的情況，使得在不同名詞或是動詞搭配時，無法正確轉換出應該對應到的字，例如：簡體字的「下面」在敘述位置時，轉換爲繁體字後爲「下面」；

而簡體字的「下面」使用在動詞時，轉換爲繁體字後則是「下麵」(李樹德，2009)。並且，簡繁用詞問題需要依靠蒐集大量的簡體以及繁體用詞的對照表來提供轉換，例如「坐公車」互相對應「坐公交車」。

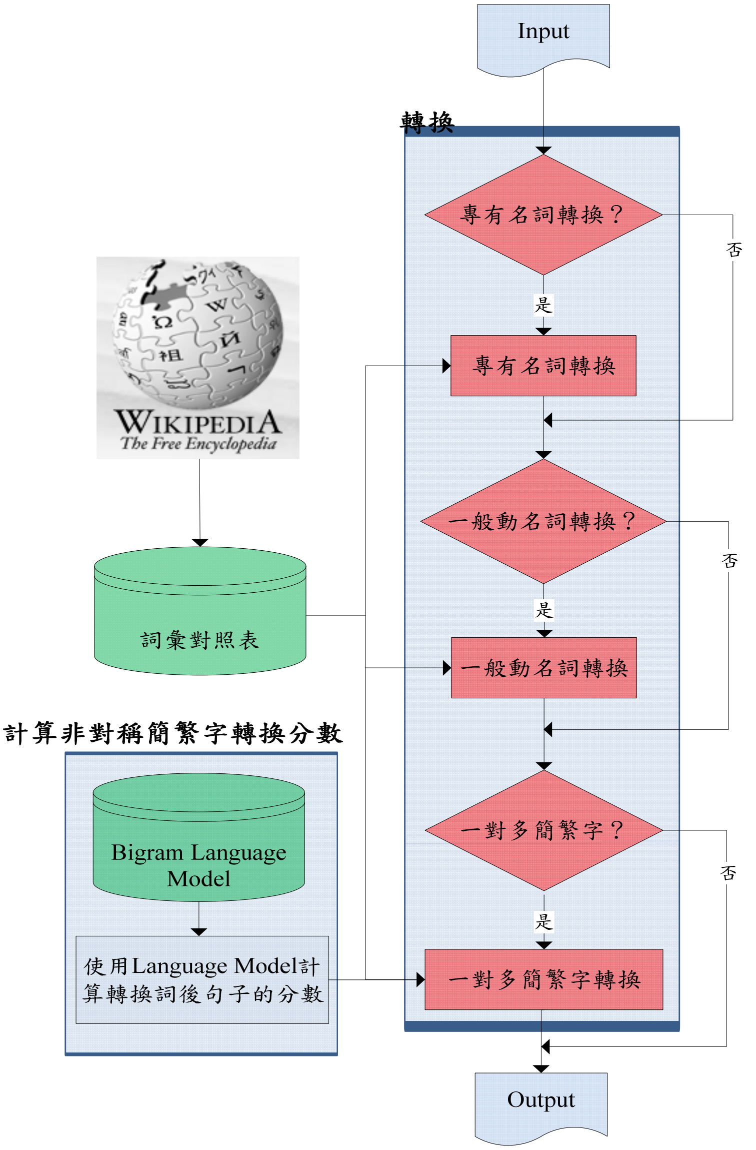
本論文提供以繁體字語料庫建構的語言模型以及收集維基百科簡繁詞彙對照表，並計算語言模型的分數來達到提升簡體字轉換繁體字正確性的方法。例如：繁體的「坐公車」轉換爲簡體的「坐公交車」，簡體的「吃面」轉換爲繁體的「吃麵」而非「吃面」。實驗部分，由於繁體字轉換簡體字爲多對一的問題，僅需查表即可完成轉換。所以我們著重於簡體字轉換繁體字時非對稱簡繁字的選擇辦法。我們以包含多種非對稱簡繁字轉換的常用字的句子進行簡體字轉換繁體字的測試，並且與目前幾種知名的翻譯工具進行比較，接著分析本系統轉換錯誤的問題。接著，我們引入斷詞系統(中科院計算所，2009)，改善原本系統無法轉換正確的幾種情況，並且分析無法正確轉換的非對稱簡繁字。最後，透過調整語言模型以及對照表的大小來驗證語言模型以及對照表大小對於簡繁轉換正確率是否有相對的影響。

2. 系統設計與方法

2.1 系統流程設計

我們根據傳統簡繁轉換系統架構做爲本系統的基礎，並加入擷取自維基百科繁體中文以及簡體中文的對應條目，利用新聞語料庫訓練 unigram 和 bigram 的機率，計算出分數最高的一對多轉換字。本篇論文的系統中，簡繁轉換的文字編碼皆使用 Unicode 的編碼方式，因爲 Unicode 爲國際編碼，它給予每一個字符唯一的編碼表示，並且包含了現有規範中所有簡體字與繁體字的日常用字。所以使用 Unicode 可以省去繁體字編碼(BIG5)與簡體字編碼(GB)的轉換步驟。轉換過程分爲三個步驟：一、首先使用對照表判斷是否需要進行專有名詞以及一般名詞轉換。二、判斷是否含有非對稱簡繁字。三、使用語言模型計算分數。

對照表內容以維基百科簡體字與繁體字條目名稱做爲專有名詞以及一般名詞的對照表、以及維基百科提供的簡繁轉換非對稱簡繁字。非對稱簡繁字爲簡體字轉繁體字一對多的狀況，例如：簡體字的「皇后」轉換爲繁體字有兩種可能，分別爲「皇后」以及「皇後」，因爲簡體字的「后」對應的繁體字爲「后」以及「後」，所以轉換上出現這兩種字詞。最後，我們蒐集 1998 年至 2001 年的新聞語料庫做爲我們的建構語言模型的語料庫，並使用語言模型計算簡體字轉換繁體字時出現非對稱簡繁字的分數。系統流程圖如圖一所示：



圖一、系統流程圖

2.2 對照表收集

中國大陸以及台灣同屬華文市場，但書籍內容用字遣詞仍有很多差異。智慧型的文體轉換，必須要解決編碼、詞彙以及簡體字對繁體字轉換時一對多及多對一的問題。兩岸不同詞彙的比對和轉換，包括人名地名組織名以及領域專有名詞(劉匯丹、吳健，2008)。目前的技術多只著重編碼的轉換，以及專有名詞的轉換。因此，對照表的內容需要豐富的轉換對照，包括成語、中外人名、地名、組織名。

對照表部份，維基百科擁有大量的條目資料，並且提供了對於一般名詞以及專業名詞的準確度，Martin Hepp (2007)提到，維基百科中有 92.67%的條目名稱即使過了一段時間後，條目名稱依然沒有改變，有 6.67%是改變了名稱，但語義上保持不變，僅剩下的0.67%爲可被刪除的條目。因此，維基百科中所有條目名稱中有 99.34%的條目名稱是可以被信任的。基於這個理由，專有名詞轉換的部份我們主要依靠維基百科做爲轉換的輸出。

維基百科提供非對稱簡繁字以及一對一單字詞的轉換對應字。因此我們以人工方式蒐集維基百科的簡體字與繁體字的非對稱簡繁字以及一對一單字詞，用來做爲簡體字與繁體字相互轉換時所依據的來源。繁體字與簡體字用詞對照部份，一共收集 7180 筆對照詞彙；非對稱簡繁字以及一對一單字詞一共收集 6619 筆，其中非對稱簡繁字一共有 475 筆，一對一單字詞爲 6144 筆。表一、表二以及表三分別爲部分對照表內容、部分非對稱簡繁字以及部分簡繁轉換一對一單字詞的範例。

表一、部分對照表內容

簡繁用詞對照表	繁體字用詞	簡體字用詞
	快閃記憶體	闪存
	網際網路	因特网
	解碼	译码
	印表機	打印机
	埠	端口
	蟻后	蚁后

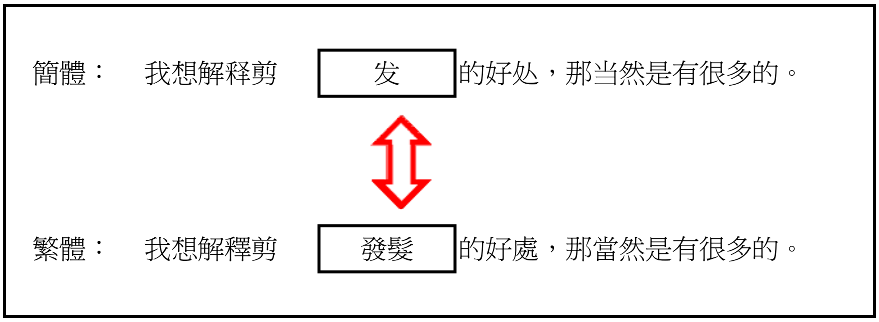
表二、部分非對稱簡繁字

非對稱簡繁字	繁體字單字詞	簡體字單字詞
	板闆	板
	辟闢	辟
	表錶	表
	發髮	发
	并並併竝	并
	乾干幹榦	干
	面麵麪麫	面

表三、部分簡繁轉換一對一單字詞

	繁體字單字詞	簡體字單字詞
簡繁轉換一對一單字詞	獃	呆
	僱	雇
	韓	韩
	號	号
	輓	挽
	兩	两
	嚴	严

如圖二所示，上面的句子爲簡體中文，下面的句子爲將轉換的繁體中文。透過對照表的簡繁用詞轉換以及一對一轉換，可以精確的轉換出正確用詞。但是，非對稱簡繁字的對照表僅提供可能轉換的字詞，並沒有提供如何正確轉換非對稱簡繁字。


簡體： 我想解释剪 发 的好处，那当然是有很多的。

繁體： 我想解釋剪 發髮 的好處，那當然是有很多的。

圖二、非對稱簡繁字的轉換問題

因此，我們使用語言模型計算非對稱簡繁字 bigram 以及 unigram 的機率值，取得圖二例子中「剪發」、「發的」以及「剪髮」、「髮的」出現機率較高的 bigram 機率值，藉由較高 bigram 機率值來做爲選擇字的轉換方式。

2.3 語言模型

我們使用統計式語言模型的方法(Statistical language model) (Rosenfeld, 1992)，篩選出正確性較高的翻譯方式(陳勇志、吳世弘、盧家慶、谷圳，2009)(洪大弘，2008)。系統使用 N-gram 語言模型計算一個句子中字詞組合的機率，機率越高代表越可能符合正確文法，反之則代表可能越不符合正確文法。首先建立語言模型，我們使用 Maximum Likelihood Estimation (MLE) (Katz, 1987)，計算出語料庫中每個字出現的相對頻率並且藉此計算機率值，如公式(1)所示：

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})} \tag{1}$$

其中，*C* 代表某個字 *W* 出現的頻率。由於一個句子是由 *n* 個字組成，因此一個句子的機率可以計算爲如公式(2)所示：

$$P(w_1^n) \equiv P(w_1, w_2, ..., w_n) \tag{2}$$

其中 w_n 表示句子中第 n 個字。 $P(w_1^n)$ 表示 1 到 n 個字出現的機率值。

假設字詞的機率爲獨立事件，一個句子條件機率可由連乘得到，如公式(3)所示：

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2) \times \ldots \times P(w_n \mid w_1^{n-1}) \\ &= P(w_1)\textstyle\prod_{k=2}^{n} P(w_k \mid w_1^{k-1)}) \end{aligned} \tag{3}$$

然而，組成一個句子的字詞是有限的，無法由過去歷史出現的無限字來做預測，因此我們將公式(3)改寫爲如公式(4)所示：

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1}) \tag{4}$$

代表依據前(n-1)個字出現的機率來預測目前第 n 個字所出現的機率，而所謂的 N-gram 就是當 *N*=2 時，稱爲 bigram，如公式(4)所示：

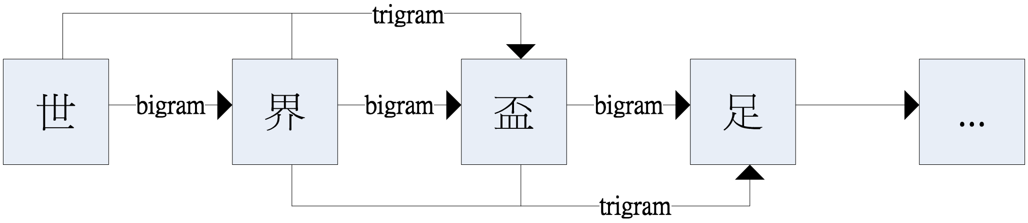
$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-1}) \tag{5}$$

當 *N*=3 時，稱爲 trigram，如公式(5)所示：

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-1}w_{n-2}) \tag{6}$$

依此類推至 N-gram。

如圖三舉例，利用前兩個字出現的情況下，預測下一個字出現的機率。如圖三所示：此圖舉例說明以「世」 與「界」爲例：由「世」出現的情況下來推測「界」出現的機率，稱爲 bigram。同理，「界」與「盃」也是 bigram；若是由「世」與「界」出現的情況下來推測「盃」出現的機率，則稱爲 trigram。然而，建構 trigram 的語言模型會造成語言模型的內容龐大，造成系統速度降低。因此，我們利用中文字出現二字詞的比例很高的特性，本研究使用的語言模型爲計算 bigram 的出現頻率。



圖三、說明 bigram 以及 trigram 計算方法

2.4 Smoothing

然而，MLE 在字詞出現頻率正常的情況下可以運作良好，但由於訓練資料稀疏，有些字詞出現的頻率會很低甚至是零，因此語言模型計算分數時可能會發生找不到要計算的字

詞，導致無法正確預測下一個字的錯誤狀況，使得正確率降低。頻率是零的情況有兩種，一種是代表兩個字之間無意義的結合，也就是真正的零；另一種是假的零，意思就是雖然這個字在文集中沒出現過，但是卻是真實世界中存在的字詞，只是訓練語料裡沒有出現。爲了避免出現機率相乘後爲零的狀況，我們使用 Smoothing 的方法，分配 bigram 以及 unigram 出現機率的權重值，以 bigram 的機率爲主要的機率分配，並給予較高的權重值；而 unigram 則給予較小的權重值，因爲 unigram 提供的線索比起 bigram 提供的線索所含有的資訊來得較低。

Smoothing的方法可分成折扣的方法和模型結合的方法，折扣的方法就是調整機率，將機率較高者分配其值給機率爲零者；而模型結合的方式就是利用內插法和補插法，當 trigram 無效時，使用 bigram，bigram 無效時則使用 unigram。本系統的 Smoothing 爲模型結合的方法。我們會使用Interpolated Kneser-Ney smoothing的演算法(Goodman, 2001)。Interpolated Kneser-Ney smoothing 公式如公式(7)所示：

$$P_{\text{int}erpolate}(w \mid w_{i-1}w_{i-2}) = \lambda P_{tirgram}(w \mid w_{i-1}w_{i-2}) + (1-\lambda)[\mu P_{bigram}(w \mid w_{i-1}) + (1-\lambda)P_{unigram}(w)] \tag{7}$$

其中 λ 以及 μ 表示分配的權重值。分別計算 trigram、bigram 以及 unigram 的機率值，並給予權重分配的 λ 和 μ ，藉以避免發生 trigram 或是 bigram 的機率值爲零的狀況。而我們主要是使用 bigram 語言模型，因此本系統使用 Interpolated Kneser-Ney smoothing 的公式時需要稍作修改，我們將公式(7)改寫爲：

$$P_{\text{int}erpolate}(w \mid w_{i-1}) = \lambda P_{bigram}(w \mid w_{i-1}) + (1-\lambda)P_{unigram} \tag{8}$$

我們刪除沒有使用到的 trigram，只計算 bigram 以及 unigram 的機率值。其中，我們認爲 bigram 的頻率的資訊強度大於 unigram 的資訊強度，因此 λ 設定爲 0.9。

2.5 Entropy以及Modified Entropy

評估的內容中有一項很重要的標準─Entropy，它被廣泛的使用在測量資訊上面(Berger, Vincent J. Della Pietra & Stephen A. Della Pietra, 1996)。其定義爲下列公式(9)：

$$H(X) = -\sum_{x \in T} P(x)\log_2 P(x) \tag{9}$$

其中隨機變數 X 涵蓋的範圍包含可預測的 T 集合(例如字母，字詞或部分的語音)，這裡表示非對稱簡繁字合併非對稱簡繁字前後單字詞的字串。 $P(x)$ 爲 MLE 所計算出來的機率值，x 表示 X 的 bigram。因爲只需要取得機率連乘後的最大值，所以我們減少公式(9)的計算量，加快計算時間。我們實際計算時使用公式(9)的 Entropy 改寫後的公式(10) Modified Entropy。

$$H^{'}(X) = -\sum_{x \in T} \log_{10} P(x) \tag{10}$$

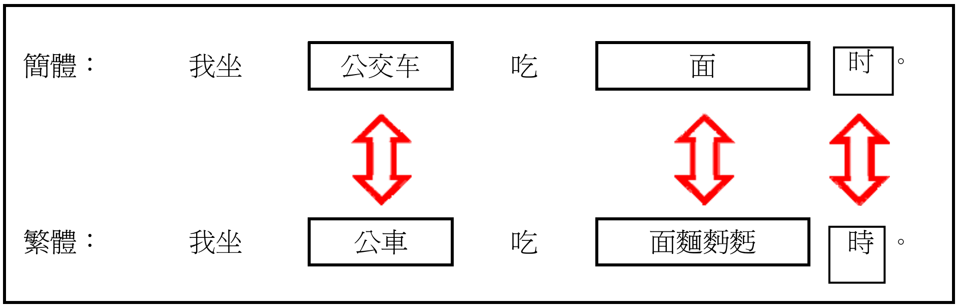
3. 實驗結果與分析

由於統計模板頻率需要大量的語料資料，因此我們蒐集新聞語料庫做爲我們的語料庫。語料庫的整理如表四所示：

表四、新聞語料庫資料明細

資料來源	年份	新聞社	文件數	檔案大小
新聞語料庫	1998-1999	China times	38,163	209MB
		China times Commercial	25,812	
		China times Express	5,747	
		Central Daily News	27,770	
		China Daily News	34,728	
	1998-1999	United Daily News	249,508	320MB
	2000-2001	United Daily News	172,421	1.03GB
		United Express	91,958	
		Ming Hseng News	168,807	
		Economic Daily News	463,873	

對照表部份，我們使用維基百科提供的非對稱簡繁字對照表，其中非對稱簡繁字的數量爲 475 筆，當系統判斷要轉換的字爲非對稱簡繁字時使用語言模型進行 bigram 的計算，選擇出分數最佳的對應字；如果系統判斷要轉換的字爲一對一單字詞時，則使用維基百科提供的 6144 筆一對一單字詞對照表直接進行轉換。系統的輸入以及輸出皆使用 Unicode 編碼的文字。我們也從維基百科中抽取了 7180 筆簡繁用詞對照表，使用方式如前述。如圖四所示：先判斷句中是否含有專有名詞以及一般動名詞，如果有則先轉換，否則進行一對一單字詞轉換以及非對稱簡繁字轉換。如圖四中出現可以對應的一般名詞的轉換，「公車」相互對應「公交车」；接著，判斷句中是否含有非對稱簡繁字，如果有則使用語言模型計算出最佳的對應字，否則直接進行轉換。如圖四中出現可以直接轉換的「時」與「时」，以及需要使用語言模型進行計算分數的「麵」與「面、麵、麪、麫」。



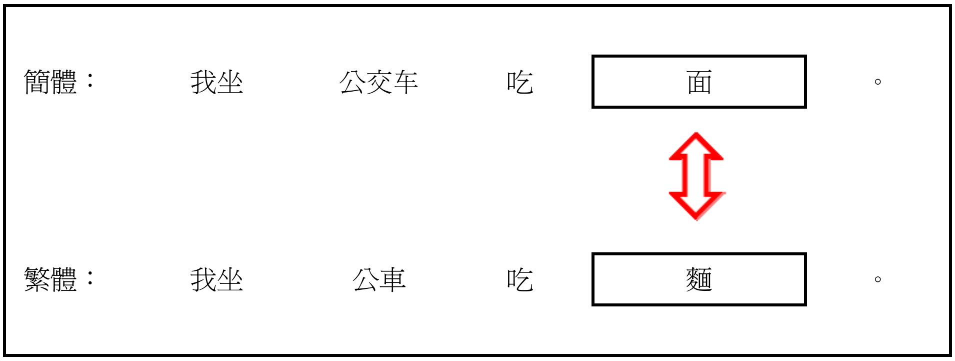
圖四、系統轉換的例子

語言模型計算分數部分，我們計算當簡體字轉換為繁體字需要選擇非對稱簡繁字時候的分數，使用前述的 Modified Entropy 做為最後計算的分數。Modified Entropy 越低代表該字的組合機率越高。因此，我們選擇計算 Modified Entropy 最低的組合。表五為圖四中非對稱簡繁字的語言模型分數計算例子。我們計算出「吃面」與「面。」、「吃麵」與「麵。」、「吃麪」與「麪。」以及「吃麫」與「麫。」等四種非對稱簡繁字組合在語言模型中出現的機率相乘的 Modified Entropy 分數。其中，「麪」以及「麫」在訓練的語料庫當中出現次數皆為 0，所以只會計算「吃」以及「。」的 unigram 機率，才會造成「吃麪。」以及「吃麪。」Modified Entropy 分數相同的狀況。

表五、非對稱簡繁字的Modified Entropy計算分數

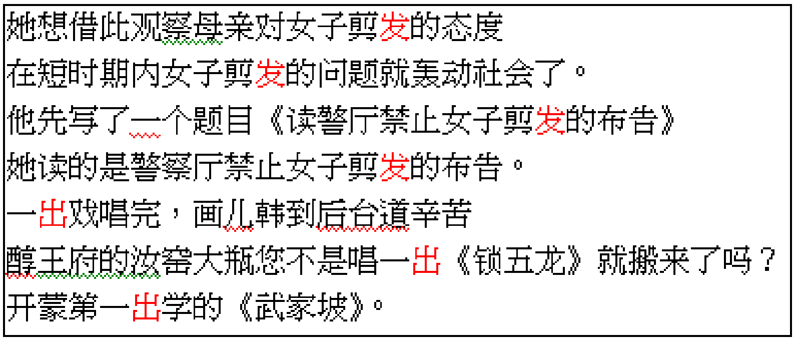
簡體=>繁體	Modified Entropy
吃面。=>吃面。	18.164046939
吃面。=>吃麵。	12.016282836
吃面。=>吃麪。	62.00000001
吃面。=>吃麫。	62.00000001

圖四最後轉換的結果如圖五所示：



圖五、轉換結果

簡繁轉換大部分的問題是出在非對稱簡繁字的問題上，因此我們的測試集主要針對句中包含非對稱簡繁字的簡體句子進行簡體轉換繁體的測試。然而，各個領域皆有其適用的簡繁轉換對照，這部份透過收集大量的對照表即可正確轉換。而在王寧(2005)中總共有 15 萬的句子，我們使用王寧(2005)提供的 271 句，其中 271 句為常見轉換字詞出現的問題包含非對稱簡繁字的簡體中文小說句子進行測試。因為小說使用的文字多為一般讀者較常接觸的一般動名詞，因此可以較準確的評估我們系統的正確性。圖六為我們進行測試的部分資料，表六為測試集的資料整理。其中，一對一單字詞為僅有一種可能的轉換結果，因此我們主要評估非對稱簡繁字轉換的正確與否。評估部分，我們使用 Accuracy，如公式(11)所定義。



她想借此观察母亲对女子剪发的态度
在短时期内女子剪发的问题就轰动社会了。
他先写了一个题目《读警厅禁止女子剪发的布告》
她读的是警察厅禁止女子剪发的布告。
一出戏唱完，画儿韩到后台道辛苦
醇王府的汝窑大瓶您不是唱一出《锁五龙》就搬来了吗？
开蒙第一出学的《武家坡》。

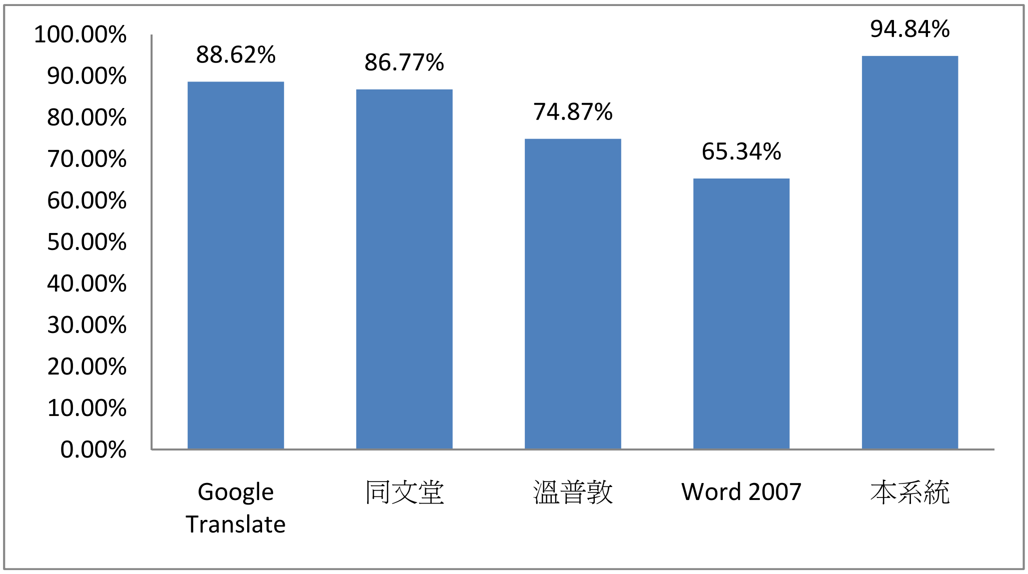
圖六、部分測試集

表六、測試集資料整理

非對稱簡繁字字數	一對一單字詞字數
756	4418

$$\text{Accuracy} = \frac{tc}{W} \times 100\% \qquad (11)$$

tc 表示正確轉換非對稱簡繁字的字數，*W* 表示非對稱簡繁字的字數。我們比較過去文獻以及目前市面上的簡繁轉換方式，並未發現同樣使用語言模型的轉換方式。目前轉換品質較佳的系統如 Google Translate、Microsoft Word 2007、溫普敦、同文堂等四種知名的翻譯軟體。圖七為這四種系統與我們系統的比較。



圖七、與其它系統比較的結果

圖七的實驗結果顯示，我們的系統對於簡繁轉換的效果比其它幾種效果來得好。因此，我們找出未被成功轉換的非對稱簡繁字，如表七所示。其中帶有底線的爲轉換錯誤的字。

表七、部份轉換錯誤的句子

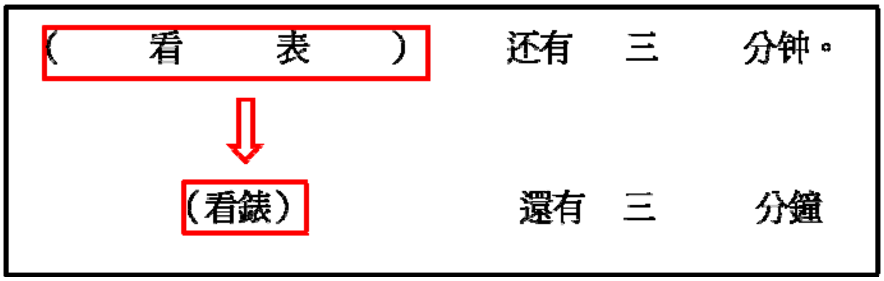
轉換錯誤的句子
已經幹了的道路
這是從前麵茶棚裡留聲機上放出來的
外麵糊了紙
現在他剛從六百裡外的煤礦回來
她摸出表來看
但是她依舊昂然自得地畫動槳
好一出大悲劇

接著，我們針對錯誤的部分，以手動方式蒐尋可能造成錯誤的對照表內容，發現對照表中含有容易因爲斷詞不佳時會造成轉換錯誤的對照詞彙，如表八所示：

表八、斷詞不佳時容易造成轉換錯誤的對照詞彙

繁體用詞	幹了	麵茶	麵糊	裡外
簡體用詞	干了	面茶	面糊	里外

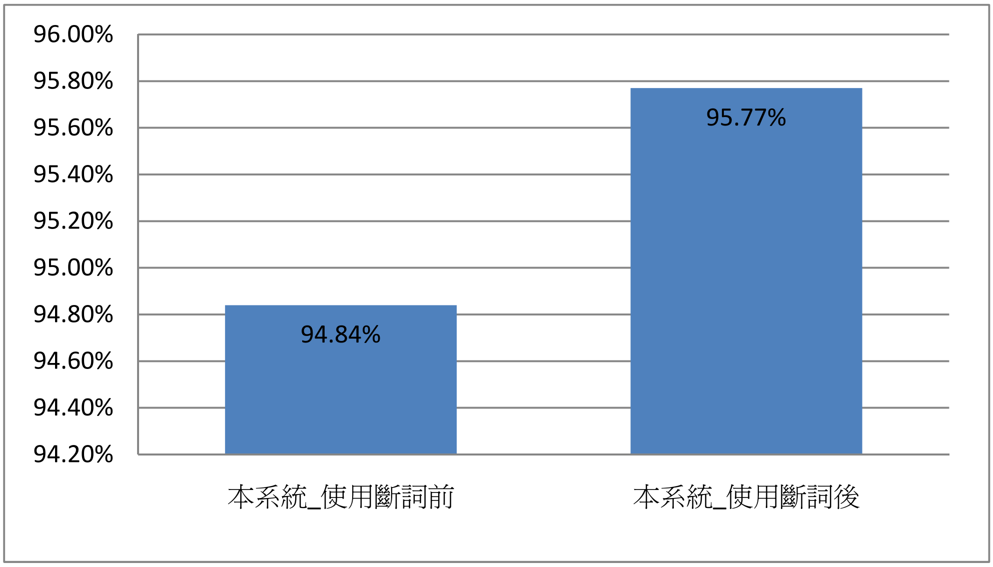
因爲它們包含了非對稱簡繁字的單字詞，因此簡體字的「干了」並非只能轉換爲如對照表中繁體字的「幹了」，而是可以轉換爲「乾了」或是「幹了」。；簡體字的「面茶」並非只能轉換爲繁體字的「麵茶」。這是因爲簡體字的「面」在繁體字時經常使用在「裡面」、「外面」、「上面」等詞彙。但是，簡體字的「面」與後一個字成詞時則成爲「麵茶」、「麵糊」等詞彙；同理，其它容易因爲斷詞不佳而造成轉換錯誤的對照表內容也是一樣的狀況。因此，斷詞的正確與否，對於簡繁轉換有著絕對的影響力。所以我們進一步引入中國科學院開發的簡體字斷詞系統(中科院計算所，2009)，嘗試改善上述的問題。我們將斷詞後不成詞的連續單字詞合併，避免文字資訊遺失。如圖八所示，經由合併連續單字詞的步驟可以保留原有的文字資訊，使得圖八例子中簡體字的「看表」可以找到對照表中繁體字的「看錶」。



圖八、合併斷詞後的連續單字詞

引入斷詞系統後，表七所示的幾種狀況可獲得改善。例如簡體字的「前面茶棚」應該爲繁體字的「前面」以及「茶棚」，但未引用斷詞的系統會因爲對照表中含有簡體字「面茶」對應至繁體字「麵茶」的關係，造成簡體字的「前面茶棚」轉換爲繁體字的「前麵茶棚」的錯誤結果；引入斷詞系統後可以正確斷出「前面」以及「茶棚」，使得轉換結果正確。再次執行實驗後，其結果如圖九所示。

實驗結果顯示，引入斷詞系統雖然可以改善系統效能，但成效不大。因此，我們關心剩下沒被成功轉換的非對稱簡繁字的類型。我們發現主要的錯誤轉換爲要轉換爲「錶」卻轉換爲「表」、要轉換爲「划」卻轉換爲「劃」、要轉換爲「齣」卻轉換爲「出」。因此，我們找出這些非對稱簡繁字被錯誤轉換時，使用語言模型計算分數的 bigram 組合字以及其句子。表九爲錯誤轉換的類型以及被轉換錯誤的 bigram 組合字的 Modified Entropy 分數，其中帶有底線的爲轉換錯誤的字。表九中錯誤轉換的非對稱簡繁字是因爲進行計算的 bigram 在語言模型的機率低於被轉換的 bigram 的機率，因而轉換爲不正確的字。然而，語言模型中所有的 bigram 皆由 unigram 組合起來。因此，unigram 頻率較高的字，自然會擁有較多的 bigram。基於這個理由，我們找出「表」、「出」、「劃」、「錶」、「齣」、「划」等六個主要被錯誤轉換的 unigram 頻率。我們發現，由於「表」、「出」、「劃」在語言模型中 unigram 的頻率皆爲「錶」、「齣」、「划」的一百倍以上，造成大多數的 bigram 皆由 unigram 頻率高的那方組成，使得「錶」、「齣」、「划」擁有較少可以依據的 bigram 頻率資訊來做爲能夠被正確轉換的 bigram 頻率。至此，我們透過對照表、語言模型以及加入斷詞後的系統，仍有無法解決的非對稱簡繁字類型，這些類型是我們認爲困難的問題。適當的頻率可以使得簡繁轉換一對多的非對稱簡繁字被正確轉換，過度的頻率會使得轉換錯誤，過度頻率只如表十所示。



圖九、第二次實驗比較系統斷詞前與系統斷詞後的結果

表九、部分錯誤轉換的類型

轉換錯誤的句子	計算的組合字	Modified Entropy 分數
醇王府的汝窯大瓶您不是唱一出《鎖五龍》就搬來了嗎？	一出《	9.949089426
	一齣《	12.5808102
開蒙第一出學的《武家坡》。	一出學	10.672644324
	一齣學	14.336685057
佩珠打算回去，她摸出表來看，快到拾二點鐘了	摸出表來	18.568075432
	摸出錶來	23.575641152
	摸齣表來	45.15278956
	摸齣錶來	69.91470532
然後自己坐到船尾，把住槳慢慢地劃起來。	地划起	14.946993213
	地劃起	13.737994911

表十、過度的頻率比較

頻率比較	出	齣
	0.004183773	0.000012075
	表	錶
	0.002347138	0.000006012
	劃	划
	0.000268397	0.000015572

我們使用語料庫的 10%(158MB)、30%(475MB)、50%(792MB)、70%(1109MB)以及 100%(1584MB)大小來建構語言模型，以及將對照表的大小分爲 10%(718 筆)、30%(2154 筆)、50%(3590 筆)、70%(5026 筆)以及 100%(7180 筆)，利用不同的語言模型大小以及對照表大小來判斷是否會影響簡繁轉換的準確性。評估方式如公式(11)定義的 Accuracy。如圖七所示：橫軸爲使用不同的語言模型大小，每一條線分別代表使用不同的對照表大小，縱軸爲 Accuracy。

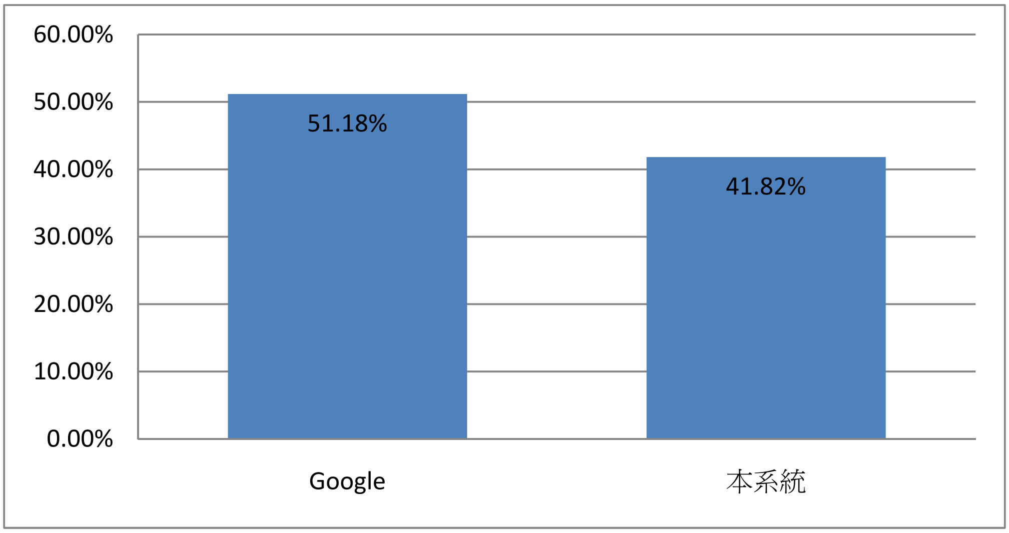
從表十一中我們可以看出，簡繁轉換的 Accuracy 隨著語言模型以及對照表使用的數量越來越大時，Accuracy 也越來越高。使用我們系統建構的 1584MB 的語言模型大小以及 7180 筆的對照表大小時 Accuracy 可達到 95.77%。其中我們注意到，當對照表大小從 50%開始，對照表對於 Accuracy 的提升較無語言模型大小 158MB 以及 475MB 時來得顯著。這是因爲我們的測試集主要針對非對稱簡繁字進行轉換的測試，大部分句子沒有包含需要簡繁用詞轉換的專有名詞以及一般名詞。劉匯丹(2008)提到，一個好的簡繁轉換系統必須要有足夠的知識，方能轉換出正確的詞彙。意思是說，因爲中國大陸與台灣因

爲文化關係，許多專有名詞以及一般動名詞使用不同名詞但是意思相同的詞彙，例如先前提到的「公車」互相對應「公交車」。因此需要大量的對照表來提供應該正確轉換的詞彙。語言模型大小部分，因爲測試集主要針對非對稱簡繁字進行轉換的測試，因此當語言模型越大時，我們可以看出 Accuracy 有顯著的提升。

表十一、簡體文字轉換繁體文字使用不同大小的語言模型以及對照表的結果

Lookup table size / Language model size	718筆	2154筆	3590筆	5026筆	7180筆
158MB	91.14%	91.80%	92.59%	92.72%	92.72%
475MB	91.93%	92.86%	93.65%	93.65%	93.92%
792MB	92.46%	92.99%	93.92%	93.92%	94.05%
1109MB	93.65%	94.05%	94.97%	94.97%	95.11%
1584MB	94.18%	94.58%	95.50%	95.50%	95.77%

繁體轉簡體的實驗部分，我們以 Google、Yahoo 網路新聞語料作爲測試資料。如圖十：包含新聞類別、科技類別以及旅遊類別一共 70 篇文章，字數一共 51350 字，其中包括了新聞文章（34 篇文章）、科技文章（26 篇文章）、旅遊文章（10 篇文章）。經由 Google 翻譯及本系統系統轉換後，一共找出 55 句裡面詞彙轉換不相同的地方。接著請以簡體字爲第一語文的使用者，選出轉換後用法較佳的詞彙，而 Google 轉換較佳詞彙有 32 個詞彙，本系統轉換後較佳詞彙有 23 個詞彙。



圖十、第三次實驗繁轉簡實驗結果

轉換結果顯示，本系統被認爲轉換不佳的詞彙的原因，在於我們使用的對照表，是 Wikipedia 提供的簡繁轉換對照表。因此，雖然我們系統成功的依照 Wikipedia 提供的對照表完成轉換，但是我們發現了兩種情況：

1. 以簡體字爲第一語文的使用者依然不習慣對照的詞彙。例如：表十二、第 6 點：公車和公交車，使用者覺得公車爲較佳，公交車並非轉換錯誤。
2. 某些詞彙需要依靠上下文來決定是否需要進行對照詞彙的轉換，或者直接轉換爲簡體字即可。如表十二、第 7 點：電腦轉換出的詞彙，前面加上 Windows，計算器在簡體用法廣義只電子產品，所以電腦爲較佳詞彙。

表十二、相同句子本系統和Google系統轉換後比較，此爲Google轉換爲較佳例子。

	Google轉換後的句子	本系統轉換後的句子
1	就是最重要的关键	就是最重要的牛鼻子
2	和我尝试的一个蓝牙设备也能够顺利地连接	和我尝试的一个蓝牙设备也能够顺利地访问
3	更欣赏他的爱家爱老婆	更欣赏他的爱家爱爱人
4	台中地检署也请相关单位搜集资料进行了解	台中地检署也请相关单位蒐集材料进行了解
5	新台币汇率	新台币外汇牌价
6	公车或火车转程就能抵达	公交车或火车转程就能抵达
7	我很容易就实现了和一台 Windows 电脑的媒体文件同步	我很容易就实现了和一台Windows计算器的媒体文件同步

另外轉換結果顯示，本系統優於 Google 在於 Google 所使用的對照表是將詞彙中的字直接做轉換。例如：表十三、第 1 點：繁體的網路在簡體用法是使用互聯網。

表十三、相同句子本系統和Google系統轉換後比較，此爲本系統轉換爲較佳例子。

	Google轉換後的句子	本系統轉換後的句子
1	MV 在网路上引发负面讨论后	MV 在互联网上引发负面讨论后
2	扫除所有垃圾资讯	扫除所有垃圾信息
3	澳洲航空(Qantas)指出	澳大利亚航天(Qantas)指出
4	并考量一般使用之超音波简易测定的数値并不精准	并考量一般使用之超声波简易测定的数値并不精准
5	除国语名称与「QQ」出奇相似外	除普通话名称与「QQ」出奇相似外
6	而且用病毒方式传播软体	而且用病毒方式传播软件
7	目前除了使用在软性可弯曲的萤幕	目前除了使用在软性可弯曲的屏幕

4. 結論

本篇論文的研究主要是改善傳統簡繁轉換僅執行一對一編碼轉換，而沒有考慮非對稱簡繁字的問題，造成非對稱簡繁字一直無法有效的被正確轉換。因此，以實驗的測試集爲例，測試集包含的非對稱簡繁字爲日常用字佔測試集中所有字數約 15%，我們的系統可以將這 15%的非對稱簡繁字執行 94.84%的正確轉換。第二次實驗加入了斷詞後僅能夠提高約 1%的正確率，並且餘下轉換錯誤的類型是我們認爲困難的轉換字，需要倚靠其它方法來解決。

實驗部分我們調整語言模型以及對照表的大小來測試是否對於簡繁轉換的效能有影響。從實驗結果來看，語言模型數量越大對於轉換結果有正向幫助，但是如果語言模型數量過大，卻會影響系統轉換的速度。這也是本系統建構語言模型時只考慮 bigram 分數的原因，因爲建構 trigram 會使得語言模型數量過大，造成轉換速度下降。對照表部分，由於中國大陸與台灣有許多用詞不同的狀況，因此需要大量的對照表提供正確轉換的詞彙。但是對照表數量過於龐大，也會造成系統轉換速度下降。因此，對照表的建構可以針對特定領域蒐集對照的詞彙，例如醫學領域的對照詞彙、資訊科學的對照詞彙…等，如此一來，針對需要轉換的用詞領域來蒐集對照表，減去不必要的資訊儲存於對照表中，避免對照表數量過大的情況。

本研究提供的方式可以讓其它研究人員以及使用者自行選擇建構語言模型的大小以及語料庫，對照表也能夠讓各人員自行選擇想要使用的對照表。因此本研究具有彈性的架構使得系統可以持續擴充改進。在未來，我們將著手建構簡體中文以及繁體中文的平行語料庫，利用簡體中文以及繁體中文的文法幾乎相同的特性，使用一些找尋新詞的方式，嘗試找出繁體中文內被判斷爲新詞的詞彙，但是簡體中文對列句子的相同位置卻沒有發現可能是名詞的詞彙，接著利用繁體中文句子中新詞的上下字爲線索，找尋出簡體中文對列句子中可能爲對應詞彙的新詞。將發現的新詞加入系統的對照表中，藉以自動擴展對照表的內容。

而在繁轉簡的實驗「沒有字轉換的錯誤」，只有「詞彙不同」的問題。例如：關鍵轉換成牛鼻子，原因在對照表關鍵定義爲牛鼻子。因此我們觀察一些不同的轉換結果，發現對照表當中有些詞彙，並不是雙向轉換都適合的，並可加入詞意的判別，增加對照表中的詞彙單向對雙向的轉換，例：當計算器前面出現 Windows 時，應轉換成电脑，利用前後文來增加在轉換的準確性。

致謝

本研究依經濟部補助財團法人資訊工業策進會「100 年度數位匯流服務開放平台技術研發計畫」辦理。

參考文獻

王曉明、魏林梅(2008，12 月)。談簡繁轉換的幾個關鍵問題。 5TH CDF 研討會數位社群雙效(CD2E)。

王寧、王曉明(2005，10 月)。兩岸四地漢字的轉換與溝通。第三屆兩岸四地中文數位化合作論壇，台北。

李樹德(2009)。Word“中文簡繁轉換”存在的問題與解決對策。2009 年 9 月 2 日，取自 http://www.yywzw.com/show.aspx?id=1570&cid=142.

劉匯丹、吳健(2008，12 月)。基於詞語消歧的分層次漢字簡繁轉換系統。5TH CDF 研討會數位社群雙效(CD2E)。

陳勇志、吳世弘、盧家慶、谷圳(2009，九月)。中文混淆字集應用於別字偵錯模板自動產生。第二十一屆自然語言與語音處理研討會，台北。

洪大弘(2008)。基於語言模型及正反面語料知識庫之中文錯別字自動偵錯系統。朝陽科技大學資訊工程系碩士論文。

Hepp, M., Siorpaes, K., & Bachlechner, D. (2007). Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management. *IEEE Internet Computing*, 11(5), 54-65.

Rosenfeld, R. (1992). *Adaptive Statistical Language Modeling: a Maximum Entropy Approach*. Ph.D. Thesis Proposal, Carnegie Mellon University..

Katz, S. (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on ACOUSTICS, SPEECH, and SIGNAL PROCESSING*, 35(3), 400-401.

Goodman, J. (2001). *A Bit of Progress in Language Modeling, Extended Version*. (Technical Report MSR-TR-2001-72). Microsoft Research, 2001.

Berger, A. L., Pietra, V. K. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 39-71.

語料庫與工具

中科院計算所 ICTCLA2009, http://ictclas.org/index.html

Google translate. http://translate.google.com.tw/#zh-CN|zh-CN|

Microsoft Office, http://office.microsoft.com/zh-tw/

滙普敦, http://www.winperturn.com.tw/

同文堂, http://tongwen.openfoundry.org/

Computational Linguistics and Chinese Language Processing
Vol. 15, No. 1, March 2010, pp. 37-60

Tourism-Related Opinion Detection and Tourist-Attraction Target Identification

Chuan-Jie Lin[*] and Pin-Hsien Chao[*]

Abstract

This paper focuses on tourism-related opinion mining, including tourism-related opinion detection and tourist-attraction target identification. The experimental data are blog articles labeled as being in the domestic tourism category in a blogspace. Annotators were asked to annotate the opinion polarity and the opinion target for every sentence. Different strategies and features have been proposed to identify opinion targets, including tourist attraction keywords, coreferential expressions, tourism-related opinion words, and a 2-level classifier. We used machine learning methods to train classifiers for tourism-related opinion mining. A retraining mechanism is proposed to obtain the system decisions of preceding sentences. The precision and recall scores of tourism-related opinion detection were 55.98% and 59.30%, respectively, and the scores of tourist attraction target identification among known tourism-related opinionated sentences were 90.06% and 89.91%, respectively. The overall precision and recall scores were 51.30% and 54.21%, respectively.

Keywords: Tourism-Related Opinion Mining, Tourist Attraction Target Identification, Opinion Analysis.

1. Introduction

The blogspace is a large resource for opinion mining. Opinion extraction methods are valuable for a wide range of applications.

Our initial interest is to extract opinions related to tourist attractions from blog articles because it is helpful to see other people’s opinions about tourist attractions when planning a tour. Nevertheless, two issues arise when trying to apply published methods to retrieve opinions of tourist attractions:

[*] Department of Computer Science and Engineering, National Taiwan Ocean University
No 2, Pei-Ning Road, Keelung, 20224 Taiwan
E-mail: {cjlin, M96570038}@ntou.edu.tw

(1) Sentence-level or document-level:

A travel article is often multi-topic because a travel route often includes several tourist attractions. Therefore, the opinion analysis for a specific tourist attraction should be carried out at sentence level, not document level.

(2) Opinion topic or opinion target:

Tourist attractions may be treated as topics (queries in IR) or as targets of opinions. Consider the following two sentences selected and adapted from our dataset:

The Dream Lake is a beautiful place.
The water is green and clear.

Both sentences are considered tourism-related opinions by us. Their opinion targets, however, are not the same. The opinion target of the first sentence is “the Dream Lake” itself, while the target of the second sentence is “the water” (in the Dream Lake). Both sentences are related to the same topic, “the Dream Lake,” but the second sentence does not contain its topic words. We find difficulty in applying the previously developed methods due to these reasons.

Opinion mining and analysis have been widely studied in several topics, including opinion detection and polarity classification (Wiebe *et al.*, 2001; Pang *et al.*, 2002; Alm *et al.*, 2005; Ghose *et al.*, 2007), opinion holder finding (Choi *et al.*, 2005; Kim & Hovy, 2005; Breck *et al.*, 2007), and opinion summarization (Ku *et al.*, 2005). Some well-known large-scale opinion mining benchmarks have also been created, such as the NTCIR MOAT datasets (Seki *et al.*, 2010) which are constructed on four languages, including Traditional Chinese.

Opinion retrieval is one of the research topics relevant to our work. Godbole *et al.* (2007) estimated the polarity scores for a large set of named entities. Nevertheless, the opinionated sentences that did not contain named entities were skipped because they measured the scores by the co-occurrences of named entities and opinion words. Ku *et al.* (2005) retrieved documents containing relevant opinions relating to TREC-like topics. Zhang *et al.* (2008) accepted short queries (titles only) and expanded the queries by web resources and relevance feedback. The units of their retrieval work, however, were documents, not sentence-level. Okamoto *et al.* (2009) extracted relevant opinionated sentences by language model. Unfortunately, a large-scale training set is required to build a reliable probabilistic model, which is labor-consuming to prepare in the tourism domain.

Opinion target identification is another research topic that is relevant to our work. Many researchers have focused on learning features of pre-defined types of products from reviews (Hu & Liu, 2004; Ghani *et al.*, 2006; Xia *et al.*, 2009). Nevertheless, the question remains whether the features of all kinds of tourist attractions are common. Moreover, in the conventional definition, an opinion target in a tourism-related opinion is not always the name of the tourist attraction.

Therefore, we define tourism-related opinion mining as a new topic and propose several approaches to solve the problem, including rule-based approaches and machine learning approaches. Although the experimental data used in this paper are written in Chinese, many of the rules and features are not language-dependent or can be easily adopted if necessary resources are available. We also hope that the experience gained from these experiments can be applied to other domains where articles are often multi-topic, such as baseball game critics.

The structure of this paper is as follows. Section 2 presents the main ideas of tourism-related opinion identification and introduces the resources prepared for the work. Section 3 describes the design of a rule-based opinion identification system. Section 4 defines the features for training classifiers to build an opinion identification system. Section 5 discusses the experimental results, and Section 6 concludes this paper.

2. Tourism-Related Opinion Analysis

2.1 Problem Definition

Opinionated sentences related to tourist attractions are the main interest of this paper. We call such an opinionated sentence a ***tourism-related opinion*** (hereafter “***TR-opinion***”) and its targeted tourist attraction a ***tourist attraction target*** (hereafter “***TA-target***”).

The main goal of this paper is to retrieve TR-opinions and determine their TA-targets. That is, given an opinionated sentence, determine whether it is tourism-related or not, and decide which tourist attraction is the focus of this opinion. Our experiments were performed based on two assumptions: (1) sentences have been *correctly* tagged as ‘opinionated’ or not; (2) tourist attraction names appearing in a document have been *correctly* recognized. Hence, we have not integrated an opinion detection module and a tourist-attraction recognition module into our system yet.

Opinion identification is not the main focus of this paper. There has been a lot of research on this topic. In the future, we would like to perform well-developed methods to do opinion detection in order to build a full system. In this paper, though, the input sentences are those sentences correctly labeled as opinions.

Tourist attraction name recognition also is not a focus of this paper. It requires a named entity recognition system specifically designed for tourist attraction names, but we cannot find one. Although some of the tourist attractions are locations or organizations, such as parks or museums, there are various types of names, such as monuments or scenic spots that would need to be learned. In this paper, we simply prepare a list of tourist attraction names and manually check the correctness of the occurrences of the attraction names in the articles.

Tourist attraction name recognition will be studied in the future.

The main ideas in accomplishing the tasks are:

(1) Some opinion words strongly hint that a sentence is tourism-related.

(2) The frequency of use of a tourist attraction and its distance to an opinionated sentence can be useful information.

(3) A tourist attraction can be expressed in several ways in an article. This is the well-known coreference problem.

(4) A sentence may target a tourist attraction if its preceding sentence also focuses on a tourist attraction.

Before designing rules or features according to these ideas, some resources were prepared beforehand, as described in the following subsections.

2.2 Experimental Dataset Preparation

The best known benchmarks for opinion mining are the NTCIR MOAT datasets (Seki *et al.*, 2010). There was one pilot task in NTCIR-6 and were two formal tasks in NTCIR-7 and NTCIR-8. There are a total of 70 topics in Traditional Chinese. Nevertheless, none of their information need is about tourism attraction opinions. Although some topics may bring in tourism-related documents, such as the terrorist bombing on Bali Island and the tsunami in Sumatra, the number of topics is too small, and we still have to find TR-opinions among the opinionated sentences. For these reasons, we decided to build a new experimental dataset in the tourism domain.

200 travel articles were collected from a blog site called Wretch[1] (無名小站). These articles were categorized as “domestic travel” on the blog site. We chose the most recommended articles by the readers in order to assure that the articles were truly about travel.

Three annotators were asked to annotate the data. Each sentence was labeled as opinionated or not, its opinion polarity was assigned, and its TA-target was found if the annotator considered it a TR-opinion.

The guidelines of TA-target decision for the annotators are as follows. Given a document, a list of tourist attractions mentioned in the document is shown to the annotators. A TA-target must be one of the tourist attractions on the list. If an opinion is made on a part of a tourist attraction (*e.g*. the souvenir shop in an amusement park), its TA-target is set to be the tourist attraction. If an opinionated sentence mentions a tourist attraction together with the city it belongs to, its TA-target is set to be the tourist attraction only. A city can be chosen as a TA-target only when the blogger directly expresses his or her feeling about the city. Note that,

[1] http://www.wretch.cc/blog

if a sentence only expresses the blogger's emotion (*e.g.* "*I am so happy today*"), it is not a TR-opinion.

The final annotations of the experimental dataset were determined by two-stage voting. The first stage determined a sentence being positive-, neutral-, negative-, or non-opinionated. The second stage determined the sentence being a TR-opinion or not by deciding its TA-target. In each stage, an option agreed upon by at least two annotators became the final annotation. If no agreement was found, the authors of this paper would choose one of the decisions made by the annotators. Those sentences voted as "non-opinionated" in the first stage were automatically labeled as "not TR-opinion" in the second stage.

Table 1. Agreements of Data Annotations

Comparison	**Opinion and Polarity**	**TR-opinion**	**TA-target**
Annotator 1 vs. 2	0.608	0.569	0.568
Annotator 1 vs. 3	0.584	0.518	0.518
Annotator 2 vs. 3	0.589	0.529	0.529
Exp Data vs. A1	0.791	0.761	0.761
Exp Data vs. A2	0.792	0.769	0.769
Exp Data vs. A3	0.758	0.701	0.701

Table 1 lists the agreement of TR-opinion and TA-target measured by Cohen's kappa coefficient. The first three rows show the agreement among the annotators. The last three rows give the agreement between the final experimental dataset and each annotator. We can see that the agreement level is not high enough. This means TR-opinion detection and TA-target identification are very challenging.

Among the 200 articles, 37 of them did not contain a tourist attraction and 7 did not contain a TR-opinion. After removing these articles, there were a total of 10,904 sentences in the remaining 156 articles, with 3,542 opinionated sentences and 1,199 TR-opinions, which leads to a precision rate of 33.9% (1199/3542) if a baseline system guesses all of the opinions as TR-opinions.

Table 2 lists the statistical data regarding the number of tourist attractions mentioned in the articles. As we can see, 28 articles contained only one tourist attraction, which means that almost 89% of the articles mentioned multiple tourist attractions, making TA-target detection an issue. There were on average 6.378 tourist attractions mentioned in each article.

Table 2. Number of Tourist Attractions in Articles

#TA	1	2	3	4	5	6	7	8	9	10	11~20	21~78	Average
#docs	28	19	23	12	13	14	9	5	6	3	17	7	6.378

2.3 Tourism-Related Opinion Words

Some opinion words are more related to tourist attractions than others. Consider the following two examples:

> *I am so excited that the vacation is coming.*
> *The lake is so large and clear.*

The adjective "excited" is often used when describing personal feelings. On the other hand, "clear" is often seen in sentences describing scenic spots. We can say that opinion words are often domain-dependent.

Many papers have focused on finding domain-specific opinion words and deciding their polarities, as mentioned in Section 1. This, however, is slightly different from our need. "Domain" in their works often refers to "a product type," such as *digital cameras*. Opinion words related to digital cameras are the adjectives used to express the features of digital cameras, such as "long" for *battery life* and "heavy" for *weight*.

Nevertheless, the question remains as to whether there are common features or attributes among tourist attractions. The feature *water* or *clearness* only relates to bodies of water, such as rivers and lakes, while the feature *design* only relates to buildings. Moreover, there are many adjectives expressing opinions directly without denoting any specific features, such as *amazing* and *beautiful* (*e.g*. "this city is beautiful"). Therefore, we want to collect a set of opinion words which are often used in tourism-related opinionated sentences without considering features.

We define a simple function ***TRscore*(*ow*)**, the *tourism-relatedness score*, to estimate the likelihood of an opinion word *ow* appearing in a TR-opinion by evaluating the ratio of the opinionated sentences where the word *ow* appears to be tourism-related:

$$TRscore(ow) = \frac{\#(ow \text{ in TR - opinion})}{\#(ow \text{ in opinion})} \tag{1}$$

Opinion words whose TR-scores are higher than a predetermined threshold are collected as the ***tourism-related opinion words*** (hereafter "***TR-opword***"). The determination of the value of the threshold of TR-scores is discussed in Section 5.1.

2.4 Coreferential Expressions

Coreference is an important problem in natural language processing. When a tourist attraction is mentioned in an article, it is quite often expressed in several different ways. Consider the following three sentences selected and adapted from our experimental dataset:

> *My family and I visited the Wufeng Resort last week.*
> *We were impressed by the fresh air when we arrived at the resort.*
> *Wufeng also thoughtfully provides parking service.*

All three underlined expressions refer to the same tourist attraction "*the Wufeng Resort*," where "*resort*" is its category, "*Wufeng*" its name, and "*the Wufeng Resort*" its full name.

It is quite common to refer a tourist attraction by the category keyword in its name. For this reason, we created a list of ***tourist attraction keywords*** (hereafter ***TA-keywords***), which are tourist attraction categories. Note that there are several synonymous keywords in the same category. The method of collecting TA-keywords is as follows.

First, a tourism website called Travel King[2] (旅遊資訊王) was visited and 1,836 tourist attraction names located in Taiwan were collected. All of the names were written in Chinese without word segmentation.

For every pair of tourist attraction names, their longest common trailing substring was extracted. The substrings containing only one Chinese character were discarded. After having humans check their correctness, 158 TA-keywords were collected, such as 國家公園 (national park) and 溫泉 (hot spring).

We do not resolve the coreference problem directly. Instead, we try to find potential coreferential expressions. The frequency or distance feature of a tourist attraction is measured by the occurrences of all kinds of coreferential expressions of this tourist attraction. The first type of coreference is expressed by the longest TA-keyword found in a tourist attraction's name.

The list of the TA-keywords may not be complete enough. Some types of names are not in the list. In order to make the system more robust, we also take the trailing substring (the last two characters) of a full name as one of its possible coreferential expressions.

Similarly, although we can extract the name part of a tourist attraction by deleting the keyword part from its full name, we simply take its leading substring (the first two characters) as one of its possible coreferential expressions.

The function ref_{all}(a) is defined to denote all possible coreferential expressions of a tourist attraction *a*. For example, ref_{all}(五峰渡假村) = {五峰渡假村, 渡假村, 五峰, 假村}, *i.e.* for the tourist attraction 五峰渡假村, its possible coreferential expressions include its full name "五峰渡假村" (*the Wufeng Resort*), its TA-keyword "渡假村" (*Resort*), its leading substring "五峰" (*Wufeng*), and its trailing substring "假村". An example of coreferential expression detection is given here:

[2] http://travel.network.com.tw/tourguide/twnmap/

我和家人上星期去五峰渡假村玩

(My family and I visited the Wufeng Resort last week.)

一到渡假村$_1$就對那邊的新鮮空氣印象深刻

(We were impressed by the fresh air when we arrived at the resort$_1$.)

五峰$_2$也貼心地提供了停車的服務

(Wufeng$_2$ also thoughtfully provides parking service.)

如果只是單純的放鬆自己什麼都不想

(If you simply want to relax and get away from it all,)

五峰渡假村是個不錯的選擇

(the Wufeng Resort will be a good choice.)

In this paragraph, a full name “*the Wufeng Resort*” (the bordered text) appears in the first and the last lines, while its TA-keyword “*resort*” (the first underlined text) is found in the second line and its leading substring “*Wufeng*” (the second underlined text) in the third line.

The strategy for finding occurrences of tourist attractions in a sentence is longest-expression-first. In other words, given a set of tourist attractions $\{A_1, A_2, \ldots, A_m\}$, we will find the attraction A_i whose coreferential expression appearing in this sentence is the longest.

This strategy has its limitations. If a tourist attraction does not reveal its category in its name, it would be difficult to know its category, such as *the Louvre* as a museum. Another limitation is to know the hierarchy of the tourist attractions. For example, some people will refer to the Wufeng Resort as a *hotel* or a *park*. How to detect a tourist attraction and identify its category will be our future work.

3. Rule-Based Approaches

To describe our approaches more clearly, Table 3 lists the definitions of notations and functions used in this paper to define opinion-mining rules and features.

The set of opinionated sentences $\boldsymbol{S_{op}}$ and the set of tourist attractions $\boldsymbol{TA}$ appearing in a document D are given in advance. Our goal is to predict a set of TR-opinions $\boldsymbol{S_{to}}$ as similar to the correct set $\boldsymbol{S^{\#}_{to}}$ as possible, and determine each TR-opinion’s TA-target. Note that we have n sentences and m tourist attractions in a document D, and $\boldsymbol{S^{\#}_{to}} \subseteq \boldsymbol{S_{op}} \subseteq \boldsymbol{S}$.

Our rule-based approaches for TR-opinion mining include the following decisions:

(1) Select a set of TR-opinion candidates $\boldsymbol{S_c}$. We can consider only a subset of the opinionated sentences $\boldsymbol{S_{op}}$ as potential TR-opinions.

(2) Select a set of TA-target candidates $\boldsymbol{TA_c}$. We can take only a subset of tourist attractions $\boldsymbol{TA}$ as TA-target candidates.

Table 3. Notations and Functions for Defining Rules and Features

Notation	**Definition**
$\boldsymbol{S}$	$\{S_1, S_2, ..., S_n\}$, the set of sentences in a document D
$\boldsymbol{TA}$	$\{A_1, A_2, ..., A_m\}$, the set of tourist attractions appearing in D
$\boldsymbol{OW}$	$\{ow_1, ow_2, ..., ow_p\}$, the set of known TR-opwords
$\boldsymbol{S_{op}}$	the set of known opinionated sentences in D
$\boldsymbol{S^{\#}_{to}}$	the set of known TR-opinions in D
$trg(s)$	the TA-target of a TR-opinion s
$freq(a)$	the frequency of a tourist attraction a, normalized by the maximal tourist attraction’s frequency in D
$\boldsymbol{A_{maxf}}$	$\arg\max_{a \in TA} freq(a)$, the set of the most frequent tourist attractions in D
$ref_{all}(a)$	the set of all possible coreferential expressions of a tourist attraction a
$in(x, j, k)$	1 if a string x appears in one of the sentences S_j, S_{j+1} ..., S_k; 0 otherwise
$fst(x, j, k)$	the index of the first sentence in S_j, S_{j+1}..., S_k which contains a string x; ∞ if none of the sentences contains x
$lst(x, j, k)$	the index of the last sentence in S_j, S_{j+1}..., S_k which contains a string x; 0 if none of the sentences contains x
$Nop_{-}(S_i)$	$\max_{k<i, S_k \in S_{op}}(k)$, the ID of the nearest opinion which precedes S_i; -1 if no preceding opinionated sentences
$Nop_{+}(S_i)$	$\min_{i<k, S_k \in S_{op}}(k)$, the ID of the nearest opinion which follows S_i; ∞ if no following opinionated sentences
$Sid_{-}(a, S_i)$	$\max_{x \in ref_c(a)} lst(x, 1, i-1)$, the ID of the nearest opinionated sentence which precedes S_i and contains a
$Sid_{+}(a, S_i)$	$\min_{x \in ref_c(a)} fst(x, i+1, n)$, the ID of the nearest opinionated sentence which follows S_i and contains a
$Nid_{-}(S_i)$	$\max_{a \in TA_c} Sid_{-}(a, S_i)$, the ID of the nearest sentence that contains a tourist attraction and precedes the sentence S_i
$Nid_{+}(S_i)$	$\min_{a \in TA_c} Sid_{+}(a, S_i)$, the ID of the nearest sentence that contains a tourist attraction and follows the sentence S_i

(3) Select a function of possible coreferential expressions $ref_c(a)$ of a tourist attraction a. We can consider only some types of expressions as coreferences to the tourist attraction a.

(4) Determine if a sentence s in $\boldsymbol{S_c}$ is a TR-opinion.

(5) Determine which tourist attraction a in $\boldsymbol{TA_c}$ is the TA-target of a TR-opinion s.

Two TR-opinion mining rules, *R*nt1 and *R*nt2, are proposed to guess a sentence S_i in $\boldsymbol{S_c}$ being a TR-opinion and its TA-target. Their definitions are explained here as illustrated in Table 4.

Nearest Preceding Tourist Attraction Rule (*R*nt1): If there is a TA-target candidate appearing inside or before S_i, it is predicted as a TR-opinion and its TA-target is the nearest tourist attraction.

Nearest in-Window Tourist Attraction Rule (Rnt2): Set the window size as b sentences. If there is a TA-target candidate appearing inside, before, or after Si in the same window, it is predicted as a TR-opinion and its TA-target is the nearest tourist attraction.

Table 4. Definitions of Base Rules

Rule	**TR-Opinion Condition**	**TA-Target**
*R*nt1	$\exists ax$, $a\in\boldsymbol{TA_c}$ and $x\in ref_c(a)$ and $lst(x, 1, i) \geq 1$	$\arg\max_{a\in TA_c, x\in ref_c(a)} lst(x,1,i)$
*R*nt2	$\exists ax$, $a\in\boldsymbol{TA_c}$ and $x\in ref_c(a)$ and $lst(x, i–b, i) \geq 1$	$\arg\max_{a\in TA_c, x\in ref_c(a)} lst(x,i-b,i)$
	$\exists ax$, $a\in\boldsymbol{TA_c}$ and $x\in ref_c(a)$ and $fst(x, i, i+b) \leq n$	$\arg\min_{a\in TA_c, x\in ref_c(a)} fst(x,i,i+b)$

The choice of $\boldsymbol{S_c}$, $\boldsymbol{TA_c}$, and $ref_c(a)$ in *R*nt1 and *R*nt2 defines different rules to detect TR-opinions and TA-targets. These settings are quickly demonstrated in Table 4 and described more clearly in the following paragraphs.

Baselines

The baseline systems use the simplest way to make the first three decisions: (1) $\boldsymbol{S_c} = \boldsymbol{S_{op}}$, *i.e.* all of the opinionated sentences are TR-opinion candidates; (2) $\boldsymbol{TA_c} = \boldsymbol{TA}$, *i.e.* all of the tourist attractions in D are TA-target candidates; and (3) $ref_c(a) = \{a\}$, *i.e.* only the full name of a tourist attraction is considered as a coreferential expression.

Table 5. Rule Settings

Rule	**Setting**
Baselines	$\boldsymbol{S_c} = \boldsymbol{S_{op}}$, $\boldsymbol{TA_c} = \boldsymbol{TA}$, $ref_c(a) = \{a\}$
*R*ow	$\boldsymbol{S_c} = \{S_i \mid S_i \in\boldsymbol{S_{op}}$ and $\exists x$, $x\in\boldsymbol{OW}$ and $in(x, i, i)=1\}$
*R*mf	$\boldsymbol{TA_c}=\boldsymbol{A_{maxf}}$
*R*cf	$ref_c(a) = ref_{all}(a)$

TR-Opword Rule (*R*ow):

In order to filter non-tourism-related sentences, such as bloggers' sentiments, an opinionated sentence is considered as a TR-opinion candidate only if it contains a TR-opword. The selection of $\boldsymbol{S_c}$ is given in the second row of Table 5.

Most Frequent Tourist Attraction Rule (*R*mf)

The most frequent tourist attraction appearing in a document *D* may be the focus of *D*. Many TR-opinions will target this tourist attraction. So, we only choose the most frequent tourist attractions in an article as the TA-target candidates, *i.e.* $\boldsymbol{TA_c}=\boldsymbol{A_{maxf}}$.

Coreferential Expression Rule (*R*cr)

All kinds of coreferential expressions, as stated in Section 2.4, are considered when determining the occurrences of a tourist attraction *a*, *i.e.* $ref_c(a) = ref_{all}(a)$.

4. Machine Learning Approach

Approaches to build a TR-opinion analysis system by machine learning are described in this section. Such a system takes a whole article (including opinions and non-opinions) as its input and returns a set of TR-opinions together with their TA-targets. Features can be divided into two sets, which are defined in Section 4.1 and Section 4.2. The options of the system's architecture and training techniques are discussed in Section 4.3 and Section 4.4.

4.1 Features for TR-Opinion Detection

The first set of features is used to detect TR-opinions, *i.e.* to determine whether an opinionated sentence S_i is tourism-related. Therefore, these features are designed for an opinionated sentence S_i. These features are quickly demonstrated in Table 6 and described more clearly in the following paragraphs.

First Sentence Feature (*f*fs)

The first sentence in an article often states the overall opinion of the author. It is interesting to see if the first sentence is tourism-related. The feature ***f*fs** finds the first sentence.

TR-Opword Features (*f*ow$_{all}$ and *f*ow$_k$)

If S_i contains a TR-opword, it is likely to be a TR-opinion. Based on this idea, two kinds of features are defined: ***f*ow$_{all}$** checks if S_i contains a TR-opword and ***f*ow$_k$** checks if S_i contains a specific TR-opword ow_k.

Table 6. Definition of TR-Opinion Detection Features

Feature	Definition of $feature(S_i)$
*f*fs	1 for S_1; 0 for other sentences in D
*f*ow$_{all}$	1 if $\exists x$, $x \in \boldsymbol{OW}$ and $in(x, i, i) = 1$; 0 otherwise
*f*ow$_k$	1 if $in(ow_k, i, i) = 1$; 0 otherwise
*f*ta$_{-1}$ / *f*tac$_{-1}$	1 if $\exists ax$, [$a \in \boldsymbol{TA}$ and $x \in ref_c(a)$ and $in(x, i–1, i–1) = 1$]; 0 otherwise
*f*ta$_0$ / *f*tac$_0$	1 if $\exists ax$, [$a \in \boldsymbol{TA}$ and $x \in ref_c(a)$ and $in(x, i, i) = 1$]; 0 otherwise
*f*ta$_{+1}$ / *f*tac$_{+1}$	1 if $\exists ax$, [$a \in \boldsymbol{TA}$ and $x \in ref_c(a)$ and $in(x, i+1, i+1) = 1$]; 0 otherwise
*f*ta$_{d-}$ / *f*tac$_{d-}$	$1 – (i–Nid_–(S_i))/n$
*f*ta$_{d+}$ / *f*tac$_{d+}$	$1 – (Nid_+(S_i)–i)/n$
*f*op$_{-1}$	1 if $Nop_–(S_i) = i–1$; 0 otherwise
*f*op$_{+1}$	1 if $Nop_+(S_i) = i+1$; 0 otherwise
*f*op$_{d-}$	$1 – (i–Nop_–(S_i))/n$
*f*op$_{d+}$	$1 – (Nop_+(S_i)–i)/n$
*f*to$_{-1}$	1 if the sentence preceding S_i is a TR-opinion; 0 otherwise
*f*to$_{d-}$	the distance score of the nearest TR-opinion preceding S_i
*f*to$^{\#}$	the 2 *f*to features whose values are assigned correctly
*f*to^2	the 2 *f*to features whose values are predicted by a retrained classifier

Tourist Attraction Distance Feature (*f*ta and *f*tac)

If an opinionated sentence is close to a tourist attraction, it is likely to be a TR-opinion and target that tourist attraction. Based on this idea, ten features are developed. The first five ***f*ta** features only consider full-name coreference, *i.e.* $ref_c(a) = \{a\}$:

***f*ta$_{-1}$**: check if the sentence preceding S_i contains a tourist attraction

***f*ta$_0$**: check if S_i contains a tourist attraction

***f*ta$_{+1}$**: check if the sentence following S_i contains a tourist attraction

***f*ta$_{d-}$**: the distance score of the nearest tourist attraction preceding S_i

***f*ta$_{d+}$**: the distance score of the nearest tourist attraction following S_i

The next five features, ***f*tac$_{-1}$**, ***f*tac$_0$**, ***f*tac$_{+1}$**, ***f*tac$_{d-}$**, ***f*tac$_{d+}$**, are defined as the same as the five ***f*ta** features, except the choice of coreference can use all kinds coreferential expressions, *i.e.* $ref_c(a) = ref_{all}(a)$.

Opinion Context Feature (*f*op)

Four features come from the surrounding opinionated sentences.

***f*op$_{-1}$**: check if the sentence preceding S_i is an opinion

***f*op$_{+1}$**: check if the sentence following S_i is an opinion

***f*op$_{d-}$**: the distance score of the nearest opinion preceding S_i

***f*op$_{d+}$**: the distance score of the nearest opinion following S_i

TR-Opinion Context Feature (*f*to)

If an opinionated sentence is close to a TR-opinion, it is likely to be tourist-related, as well. Two features are introduced here:

***f*to$_{-1}$**: the sentence preceding S_i is a TR-opinion

***f*to$_{d-}$**: the distance score of the nearest TR-opinion preceding S_i

Note that we do not know the values of these two features for a new article (nor should we when testing on the test set). In such a case, both feature values of the first sentence are set to be 0 because there is no preceding sentence. The predicted result of a sentence will be used to determine the two feature values of its following sentence. More ideas about these features are discussed in Section 4.4.

4.2 Features for TR-Target Identification

The second set of features is used to identify TA-targets, *i.e.* to determine whether a tourist attraction A_j is the TA-target of an opinionated sentence S_i. Therefore, these features are designed for a pair of $<S_i, A_j>$ given an opinionated sentence S_i and a tourist attraction A_j. These features are quickly demonstrated in Table 7 and described more clearly in the following paragraphs. The candidates of TA-targets are the set of tourist attractions appearing in the article.

Frequency Feature (*f*fq)

Similar to the idea of the Most-Frequent-Tourist-Attraction Rule, the occurrence of a tourist attraction is taken into account.

Table 7. Definition of TR-Opinion Detection Features

Feature	Definition of $feature(S_i, A_j)$
*f*fq	$freq(A_j)$
*f*na$_{n-}$ / *f*nac$_{n-}$	1 if $Nta_{-}(S_i) = A_j$; 0 otherwise
*f*na$_{n+}$ / *f*nac$_{n+}$	1 if $Nta_{+}(S_i) = A_j$; 0 otherwise
*f*na$_{d-}$ / *f*nac$_{d-}$	$1 - (i - Sid_{-}(A_j, S_i))/n$
*f*na$_{d+}$ / *f*nac$_{d+}$	$1 - (Sid_{+}(A_j, S_i) - i)/n$

Distance Feature (*f*na and *f*nac)

It is intuitive that a TR-opinion is often close to its targeting tourist attraction. Eight features are derived from the distance of an opinionated sentence S_i and a tourist attraction A_j. The first four ***f*na** features only consider full-name coreference, *i.e.* $ref_c(a) = \{a\}$:

***f*na**$_{\mathbf{n-}}$: check if A_j is the nearest tourist attraction preceding S_i

***f*na**$_{\mathbf{n+}}$: check if A_j is the nearest tourist attraction following S_i

***f*na**$_{\mathbf{d-}}$: the distance score of A_j and S_i when A_j precedes S_i

***f*na**$_{\mathbf{d+}}$: the distance score of A_j and S_i when A_j follows S_i

The next four features, ***f*nac**$_{\mathbf{n-}}$, ***f*nac**$_{\mathbf{n+}}$, ***f*nac**$_{\mathbf{d-}}$, ***f*nac**$_{\mathbf{d+}}$, are defined as the same as the four ***f*na** features, except the choice of coreference can use all kinds coreferential expressions, *i.e.* $ref_c(a) = ref_{all}(a)$.

4.3 Retraining by Prediction

The TR-Opinion Context Feature (***f*to**) is very useful but also dangerous. We conducted an oracle model where the values of the TR-Opinion Context Feature of the test data were set correctly (denoted as ***f*to**$^{\#}$), and found that the performance was the best (as depicted later in Table 10). Nevertheless, if the feature values came from the predictions of the classifier, the errors would propagate and harm the performance greatly (also depicted in Table 10).

We propose a retraining method to use the TR-Opinion Context Feature. Training is performed in three steps. First, set the values of the TR-Opinion Context Feature of the training data correctly to train a preliminary classifier. Use this preliminary classifier to predict the TR-opinions in the training set. Then, use the predictions to assign the values of the TR-Opinion Context Feature of the training data to train a classifier. The second classifier is used to construct the real TA-target identification system. The values of the TR-Opinion Context Feature predicted by the second classifier are denoted as ***f*to**$^{\mathbf{2}}$.

4.4 Single-Layer and Dual-Layer Models

Our TA-target identification system is constructed as follows: each sentence in an article is paired with each of the tourist attractions appearing in the article and labeled by a classifier. If none of the pairs is classified as positive, this sentence is not a TR-opinion. Otherwise, the sentence is predicted as a TR-opinion and all the tourist attractions in the pairs receiving positive predictions are its TR-targets.

The process of TA-target identification can be divided into two steps: detecting TR-opinions and assigning TR-targets to them. Hence, we can train two classifiers for the two steps separately, or train a single classifier to identify the TA-targets directly. Two different models are designed, given that the input is a pair of an opinionated sentence S_i and a tourist attraction A_j.

Single-Layer Model

The classifier directly determines whether the tourist attraction A_j is the TR-target of the sentence S_i. All of the features introduced in Section 4.1 and 4.2 are used for training even if a feature only relates to the sentence S_i only.

Dual-Layer Model

The classification module consists of two classifiers. The first-layer classifier determines whether S_i is a TR-opinion. Only features introduced in Section 4.1 are used to train the first-layer classifier. If S_i is classified as a TR-opinion, the pair $<S_i, A_j>$ is passed to the second-layer classifier. The second-layer classifier determines whether A_j is the TR-target of S_i. Only features introduced in Section 4.2 are used to train the second-layer classifier.

5. Experiments

The experiments shown in this section were all conducted in a leave-one-out cross-validation fashion where each of the 156 articles in the experimental data set was kept out as the test data and the others as the training data in turn.

The number of the positive examples is relatively small compared to the negative examples. We did not evaluate the system by accuracy because the majority prefers guessing all sentences as “not TR-opinion”. Additionally, in order to create a balanced training set, we randomly selected negative examples in the same amount of the positive examples in each training set.

Both TR-opinion detection and TA-target identification are evaluated by the micro-average precision (P), recall (R), and F-measure (F), where $F = \frac{2 \times P \times R}{P + R}$.

For TR-opinion detection,

$$P = \frac{\#(\text{correctly guessed TR - opinions})}{\#(\text{TR - opinions guessed by system})} \quad (2)$$

$$R = \frac{\#(\text{correctly guessed TR - opinions})}{\#(\text{real TR - opinions})} \quad (3)$$

For TA-target identification,

$$P = \frac{\#(\text{correctly guessed TA - targets})}{\#(\text{TA - targets guessed by system})} \quad (4)$$

$$R = \frac{\#(\text{correctly guessed TA - targets})}{\#(\text{real TA - targets})} \quad (5)$$

5.1 Tourism-Related Opinion Word Selection

As introduced in Section 2.3, we want to find opinion words highly related to tourism. A preliminary experiment was conducted to determine the threshold of TR-scores to select TR-opwords. The candidates of TR-opwords were the opinion words collected in NTUSD, the National Taiwan University Sentiment Dictionary (Ku & Chen, 2007).

The threshold of the TR-scores was determined by the baseline experiment of TR-opinion detection. Set the threshold values varying from 0 to 1 with a step of 0.01 and selected those opinion words whose TR-scores were higher than the threshold to predict TR-opinions by the TR-Opword Rule only.

Table 8. Performance of TR-Opinion Detection under Different Thresholds

Threshold	#TR-ow	P	R	F
0	482.1	37.71	46.46	41.63
0.1	475.2	38.71	46.04	42.06
0.2	443.5	41.42	43.29	42.33
0.25	**418.6**	**43.17**	**41.62**	**42.38**
0.26	**418.6**	**43.17**	**41.62**	**42.38**
0.3	408.8	42.82	39.78	41.25
0.4	359.7	46.58	31.78	37.78
0.5	266.2	49.28	22.77	31.15
0.6	251.3	50.23	18.18	26.70
0.7	218.4	49.06	10.93	17.87
0.8	202.5	50.50	8.42	14.44

Table 8 shows the results of TR-opinion detection under different threshold settings. The threshold value achieving the best performance was 0.25 and 0.26, but not significantly the best if compared to a nearby setting. We chose 0.25 as the threshold in the following experiments. Note that the sets of TR-opwords were not the same in different iterations of cross-validation because the training sets were different. The second column of Table 8 depicts the average number of TR-opwords selected in each iteration.

5.2 Experiments of Rule-Based Approaches

Table 9 presents the results of the rule-based TA-target identification systems under different rule combinations. The Nearest-TA-in-Window Rule (*R*nt2) slightly outperformed the Nearest- Preceding-TA Rule (*R*nt1) in any combination. The rule combination achieving the best performance was the Nearest-TA-in-Window Rule (*R*nt2) combined with the Coreferential Expression Rule (*R*cr), which was significantly different from all the others.

Table 9. Performance of the Rule-Based TA-Target Identification Systems

Rule Combination	**P**	**R**	**F**
*R*nt1	25.74	70.73	37.74
*R*nt1+*R*ow	32.21	29.44	30.76
*R*nt1+*R*mf	18.84	46.96	26.89
*R*nt1+*R*cr	27.01	74.65	39.67
*R*nt1+*R*ow+*R*cr	19.16	47.79	27.35
*R*nt1+*R*mf+*R*cr	34.18	31.28	32.67
*R*nt1+*R*ow+*R*mf+*R*cr	23.16	19.43	21.13
*R*nt2 (*b*=5)	29.93	52.54	38.14
*R*nt2+*R*ow	35.21	21.93	27.03
*R*nt2+*R*mf	22.90	26.61	24.61
***R*nt2+*R*cr**	**32.10**	**60.88**	**42.04**
*R*nt2+*R*ow+*R*cr	25.34	31.53	28.09
*R*nt2+*R*mf+*R*cr	37.47	25.19	30.12
*R*nt2+*R*ow+*R*mf+*R*cr	28.46	12.68	17.54

5.3 Experiments of Machine Learning Approaches

We used the LIBSVM tool (Fan *et al*., 2005) to train the classifiers. We chose SVM because some features' domains were sets of real numbers, not strings.

The dual-layer model first detects the TR-opinions then identifies the TA-targets. We evaluated the first-layer (for TR-opinion detection) and second-layer (for TA-target identification) classifiers separately.

5.3.1 TR-Opinion Detection Experiments

Table 10 presents the selected results of TR-opinion detection by different combinations of features where *f*xx$_{-}$ denotes all *f*xx features regarding objects preceding the sentence (*i.e*. *f*xx$_{-1}$ and *f*xx$_{d-}$), and *f*xx$_{0-}$ denotes the feature combination of *f*xx$_{-}$ and *f*xx$_{0}$.

The results in Table 10 are represented in groups. The experiments in the first group only used the Tourist Attraction Distance Features (*f*ta). The feature combinations in the second group were suggested by a feature selection method, WLLR, which will be introduced later.

Table 10. Results of the TR-Opinion Detection by Machine Learning, Rules, and Annotators

Feature Combination	**P**	**R**	**F**
$f\mathrm{ta}$	42.15	60.88	49.81
$f\mathrm{ta}_{-}$	40.92	80.23	54.20
$f\mathrm{ta}_{0-}$	61.18	36.28	45.55
$f\mathrm{tac}$	56.90	47.79	51.95
$f\mathrm{tac}_{-}$	41.95	84.07	55.97
$f\mathrm{tac}_{0-}$	62.28	44.20	51.71
$f\mathrm{ow}_{all}+f\mathrm{tac}+f\mathrm{to}^2$	55.67	58.97	57.27
$f\mathrm{ow}_{all}+f\mathrm{tac}_{0-}+f\mathrm{to}^2$	54.91	60.13	57.40
$f\mathrm{ow}_{all}+f\mathrm{fs}+f\mathrm{op}_{-}+f\mathrm{tac}_{-}+f\mathrm{to}^2$	48.48	61.38	54.18
$f\mathrm{ow}_{all}+f\mathrm{fs}+f\mathrm{op}_{-}+f\mathrm{tac}_{0-}+f\mathrm{to}^2$	54.34	58.97	56.56
$\boldsymbol{f\mathbf{ow}_{all}+f\mathbf{fs}+f\mathbf{op}+f\mathbf{tac}+f\mathbf{to}^2}$	**55.98**	**59.30**	**57.59**
$f\mathrm{ow}_{all}+f\mathrm{fs}+f\mathrm{op}+f\mathrm{ta}+f\mathrm{to}^2$	50.68	53.13	51.87
$f\mathrm{ow}_{all}+f\mathrm{fs}+f\mathrm{op}_{-}+f\mathrm{to}^{\#}$	58.77	79.40	67.54
$f\mathrm{ow}_{all}+f\mathrm{fs}+f\mathrm{op}_{-}+f\mathrm{tac}_{-}+f\mathrm{to}^{\#}$	65.37	64.22	64.79
$f\mathrm{ow}_{all}+f\mathrm{fs}+f\mathrm{op}_{-}+f\mathrm{tac}_{-}+f\mathrm{to}$	57.60	40.12	47.30
$R\mathrm{nt2}+R\mathrm{cr}$	43.14	81.82	56.49
Annotator 1	85.62	88.91	87.23
Annotator 2	89.17	82.40	85.65
Annotator 3	96.52	57.80	72.30

The experiments in the third and the fourth groups tried more feature combinations but used the TR-opinion Context Features in different ways. The fourth group used the TR-opinion Context Feature after Retraining ($f\mathrm{to}^2$). The fourth group used correct values for the TR-opinion Context Features ($f\mathrm{to}^{\#}$, as oracle model) and prediction by the previously trained model without retraining ($f$to).

The fifth one has the best performance achieved by the rule-based model and the final group lists the performances of human annotators which can be regarded as upper bounds.

The second and the third groups of results show that the TR-opinion Context Feature after Retraining ($f\mathrm{to}^2$) is useful, for the best performances were achieved by those feature combinations containing $f\mathrm{to}^2$. Compared with the fourth group, the oracle model (containing *f*to$^{\#}$) outperforms other combinations, which concludes that *f*to$^{\#}$ is a great feature but, unfortunately, is unattainable. On the other hand, using the prediction by the classifier without retraining (*f*to) harmed the performance. We can say that the retraining process did improve the performance.

The first group also suggests that the Preceding Tourist Attraction Distance Features with or without Coreferential Expressions (*f*ta$_{-}$ and *f*tac$_{-}$) are useful.

To see the usefulness of features, we used an adapted version of WLLR (Weighted Log Likelihood Ratio) (Nigam *et al*., 2000) to measure the usefulness of the features. The adapted equation of WLLR in our work is:

$$WLLR(f) = \underset{x \in P}{avg}(f(x)) \log \frac{\underset{x \in P}{avg}(f(x))}{\underset{x \in N}{avg}(f(x))} \tag{6}$$

Table 11. WLLR of Features

Feature	***avg$_P$(f)***	***avg$_P$(f)* / *avg$_N$(f)***	**WLLR**
*f*to$^{\#}_{-1}$	0.371	8.204	0.781
*f*tac$_0$	0.272	5.588	0.468
*f*to$^{\#}_{d-}$	0.853	1.599	0.401
*f*ta$_0$	0.220	5.930	0.392
*f*tac$_{-1}$	0.258	2.614	0.248
*f*tac$_{d+}$	0.832	1.280	0.205
*f*ta$_{-1}$	0.210	2.438	0.187
*f*ta$_{d+}$	0.788	1.259	0.181
*f*ow$_{all}$	0.416	1.484	0.164
*f*tac$_{d-}$	0.903	1.198	0.163
*f*ta$_{d-}$	0.875	1.185	0.148
*f*tac$_{+1}$	0.192	1.677	0.099
*f*ta$_{+1}$	0.160	1.638	0.079
*f*op$_{d+}$	0.938	1.028	0.026
*f*op$_{d-}$	0.931	1.017	0.015
*f*op$_{-1}$	0.463	1.033	0.015
*f*op$_{+1}$	0.460	1.022	0.010
*f*fs	0.038	0.817	-0.008

where *f*(*x*) is a feature function which defines a numerical feature value for a given example *x*, *avg*(***v***) means the average over a numerical set ***v***, *P* and *N* are the sets of positive examples and negative examples in the training set, respectively. The adaptation is made to make it applicable for both Boolean features (treated as 0 and 1) and numerical features.

Table 11 lists the WLLR and averages (over positive and negative examples) of the features. As we can see, the best features according to WLLR are the TR-Opinion Context Features (***f*to**), the Tourist Attraction Distance Features (***f*ta** and ***f*tac**, with or without coreferential expressions), and the All-TR-Opword Feature ($\boldsymbol{f}\mathbf{ow}_{\mathbf{all}}$). The experiments inspired by feature selection are listed in the second group. The results in Table 10 support the predictions by WLLR as the feature combination $f\text{ow}_{all}+f\text{tac}_{0}+f\text{to}^{2}$ performs very well.

The best performance, however, where an F-measure score of 57.59% is achieved, is by the feature combination using all kinds of features. It outperforms the combination by feature selection significantly ($p<0.001$).

5.3.2 TA-Target Identification Experiments

Table 12 lists the experimental results of TA-target identification by different approaches. The second row gives the performance of the second-layer classifier where the first-layer was replaced by a perfect model, *i.e.* only known TR-opinions were assigned TA-targets. The precision and recall scores were 90.06% and 89.91%, respectively, and the F-measure score was around 90%. This means that the bottleneck of this work is TR-opinion detection. The third row shows the performance of the overall dual-layer system consisting of the best models of the two layers, which F-measure is 52.72% and is the best among all TA-target identification models.

The models of the fourth and the fifth rows are single-layer classifiers. Even when the correct values of TR-Opinion Context Features ($f\text{to}^{\#}$) are used, they still cannot compete with the dual-layer model. This shows that dual-layer classification is a better approach.

The sixth row of Table 12 gives the performance of TA-target identification by rules. Although the best rule-based approach performs well in TR-opinion detection, its ability to identify TA-targets is weaker.

The last three rows present the performance of the results of the three annotators. We can see that the best F-measure of a ML-based system is about 60% to 75% of human ability. So, there is still room to improve.

Table 12. Results of TA-Target Identification by Different Approaches

Feature Combination	**P**	**R**	**F**
The second layer only (TA-Target Identification)			
*f*fq+*f*nac	90.06	89.91	89.98
Dual-Layer Model			
1st layer: *f*ow$_{all}$+*f*fs+*f*op+*f*tac+*f*to^{2} 2nd layer: *f*fq+*f*nac	**51.30**	**54.21**	**52.72**
Single-Layer Model			
*f*ow$_{all}$+*f*fs+*f*op$_{-}$+*f*to$^{\#}$+*f*fq+*f*nac	32.83	88.91	47.95
*f*ow$_{all}$+*f*fs+*f*op$_{-}$+*f*tac+*f*to$^{\#}$+*f*fq+*f*nac	32.75	88.74	47.84
*R*nt2+*R*cr	32.10	60.88	42.04
Annotator 1	84.10	87.32	85.68
Annotator 2	87.27	80.65	83.83
Annotator 3	94.71	56.71	70.94

6. Conclusions and Future Work

This paper aims at detecting tourism-related opinionated sentences and identifying their tourist attraction targets. Several rules and features were invented and tested in different combinations. The performance is improved by building a dual-layer classification system where the classifiers of TR-opinion detection and TA-target identification are trained separately. Retraining by the prediction method is introduced to decide the values of the TR-Opinion Context Features. This feature, together with the tourism-related opinion words and distances to the tourist attractions were verified to be useful. The best overall performance of TA-target identification is 52.72%, which is about 60% to 75% of human ability.

In the future, we would like to implement known methods to do opinion detection and tourist attraction recognition so we can build a real system and evaluate its performance. More features should be studied for TR-opinion detection.

By the location information of the tourist attractions, it is also interesting to make a summary for a city or a country by the opinions about the tourist attractions located in that area. This will be our future work.

Reference

Alm, C. O., Roth, D., & Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceeding of HLT/EMNLP 2005*, 579-586.

Breck, E., Choi, Y., & Cardie, C. (2007). Identifying Expressions of Opinion in Context. In *Proceeding of IJCAI 2007*, 2683-2688.

Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S. (2005). Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In *Proceeding of HLT/EMNLP 2005*, 355-362.

Fan, R.E., Chen, P.H., Lin, C.J., & Joachims, T. (2005). Working Set Selection Using the Second Order Information for Training SVM. In *Journal of Machine Learning Research*, 6, 1889-1918.

Ghani, R., Probst, K., Liu, Y., Krema, M., & Fano, A. (2006). Text Mining for Product Attribute Extraction. In *SIGKDD Explorations*, 1(8), 41-48.

Ghose, A., Ipeirotis, P., & Sundararajan, A. (2007). Opinion Mining using Econometrics: A Case Study on Reputation. In *Proceeding of ACL 2007*, 416-423.

Godbole, N., Srinivasaiah, M., & Skiena, S. (2007). LargeScale Sentiment Analysis for News and Blogs. In *Proceedings of ICWSM 2007*, 219-222.

Hu M. & Liu, B. (2004). Mining Opinion Features in Customer Reviews. In *Proceeding of AAAI 2004*, 755-760.

Kim, S.M., & Hovy, E. (2005). Identifying Opinion Holders for Question Answering in Opinion Texts. In *Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains*, 1-8.

Ku, L.W. & Chen, H.H. (2007). Mining Opinions from the Web: Beyond Relevance Retrieval. In *Journal of American Society for Information Science and Technology*, Special Issue on Mining Web Resources for Enhancing Information Retrieval, 58(12), 1838-1850.

Ku, L.W., Lee, L.Y., Wu, T.H., & Chen, H.H. (2005). Major Topic Detection and Its Application to Opinion Summarization. In *Proceedings of SIGIR 2005*, 627-628.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents Using EM. In *Machine Learning*, 39(2-3), 103-134.

Okamoto, T., Honda, T., & Eguchi, K. (2009). Locally Contextualized Smoothing of Language Models for Sentiment Sentence Retrieval. In *Proceeding of the 1st international CIKM 2009 Workshop on Topic-Sentiment Analysis for Mass Opinion*, 73-80.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proceeding of EMNLP 2002*, 79-86.

Seki, Y., Ku, L.W., Sun, L., Chen, H.H., & Kando, N. (2010). Overview of Multilingual Opinion Analysis Task at NTCIR-8: A Step Toward Cross Lingual Opinion Analysis. In *Proceedings of NTCIR-8*, 209-220.

Wiebe, J., Wilson, T., & Bell, M. (2001). Identifying Collocations for Recognizing Opinions. In *Proceeding of ACL 2001 Workshop on Collocation*, 24-31.

Xia, Y., Hao, B., & Wong, K.F. (2009). Opinion Target Network and Bootstrapping Method for Chinese Opinion Target Extraction. In *Lecture Notes in Computer Science*, 5839, 339-350.

Zhang, W., Jia, L., Yu, C., & Meng, W. (2008). Improve the Effectiveness of the Opinion Retrieval and Opinion Polarity Classification. In *Proceedings of the CIKM 2008*, 1415-1416.

Computational Linguistics and Chinese Language Processing
Vol. 15, No. 1, March 2010, pp. 61-76

Cross-Validation and Minimum Generation Error based Decision Tree Pruning for HMM-based Speech Synthesis

Heng Lu[*], Zhen-Hua Ling[*], Li-Rong Dai[*], and Ren-Hua Wang[*]

Abstract

This paper presents a decision tree pruning method for the model clustering of HMM-based parametric speech synthesis by cross-validation (CV) under the minimum generation error (MGE) criterion. Decision-tree-based model clustering is an important component in the training process of an HMM based speech synthesis system. Conventionally, the maximum likelihood (ML) criterion is employed to choose the optimal contextual question from the question set for each tree node split and the minimum description length (MDL) principle is introduced as the stopping criterion to prevent building overly large tree models. Nevertheless, the MDL criterion is derived based on an asymptotic assumption and is problematic in theory when the size of the training data set is not large enough. Besides, inconsistency exists between the MDL criterion and the aim of speech synthesis. Therefore, a minimum cross generation error (MCGE) based decision tree pruning method for HMM-based speech synthesis is proposed in this paper. The initial decision tree is trained by MDL clustering with a factor estimated using the MCGE criterion by cross-validation. Then the decision tree size is tuned by backing-off or splitting each leaf node iteratively to minimize a cross generation error, which is defined to present the sum of generation errors calculated for all training sentences using cross-validation. Objective and subjective evaluation results show that the proposed method outperforms the conventional MDL-based model clustering method significantly.

Keywords: Speech Synthesis, Hidden Markov Model, Decision Tree Pruning, Cross-validation, Minimum Generation Error.

[*] University of Science and Technology of China, No. 96, Jinzhai Road, Hefei, Anhui, China
Tel: (+86) 13721053256; fax: (+86) 551 5331801.
E-mail: luhenglh@mail.ustc.edu.cn; zhling@ustc.edu; lrdai@ustc.edu.cn; rhw@ustc.edu.cn
The author for correspondence is Heng Lu.

1. Introduction

Currently, there are two main speech synthesis methods. One is unit-selection speech synthesis (Hunt & Black, 1996) (Ling & Wang, 2007) and the other is the hidden Markov model (HMM) based parametric speech synthesis (Black, Zen, & Tokuda, 2007). The unit-selection approach concatenates the natural speech segments selected from a recorded database to produce synthetic speech. It can generate highly natural speech often, but its performance may degrade severely when the contexts for synthesis are not included in the database. In HMM-based parametric speech synthesis, speech waveforms are parameterized and modeled by HMMs in model training (Yoshimura, Tokuda, Masuko, Kobayashi, & Kitamura, 1999). During synthesis, speech parameters are generated from the trained models (Tokuda, Yoshimura, Masuko, Kobayashi, & Kitamura, 2000) and sent to a parametric synthesizer to reconstruct speech waveforms. Although the quality of synthetic speech still needs improvement, HMM-based parametric synthesis has several important advantages, including high flexibility of the statistical models, a comparatively small database necessary for system construction and robust performance of the synthetic speech -- it never makes the serious errors that unit-selection speech synthesis may make sometimes.

In HMM-based parametric speech synthesis, binary decision tree based context-dependent model clustering is a necessary step in dealing with data-sparsity problems and predicting model parameters for the contextual features of synthetic speech that do not occur in the training set. In the conventional model clustering process, the maximum likelihood (ML) criterion is utilized to choose the optimal question from the question set for each tree node split and the minimum description length (MDL) criterion (Shinoda & Watanabe, 2000) is used as the stopping criterion to control the size of trained decision trees, which affects the performance of synthetic speech significantly, *e.g.*, a large decision tree may alleviate the over-smoothing effects in generated speech parameters but may also lead to over-fitting problems. Nevertheless, the MDL criterion is derived based on an asymptotic assumption and the assumption that fails when there is not enough training data (Rissanen, 1980). Therefore, it may not work successfully in HMM-based speech synthesis, where the amount of training data is much smaller than that in speech recognition.

Some research work has been done to improve the MDL criterion for the decision tree construction of HMM-based speech synthesis. A decision tree backing-off method was proposed in (Kataoka, Mizutani, Tokuda & Kitamura, 2004). In this method, a decision tree was first built using ML criterion without pruning. During synthesis, the tree nodes that generated the observations with maximum likelihood were chosen by a process of backing-off from the leaf node that was decided by the contextual information of each state for synthesis to the root node. Nevertheless, there still exist two issues in this method. One is the one-dimensional optimization algorithm adopted in (Kataoka, Mizutani, Tokuda, & Kitamura,

2004) to reduce the computational complexity, which means the decision tree backing-off is conducted simultaneously for all states instead of processing each state separately. The other is the inconsistency between the ML criterion and the aim of speech synthesis, which is to generate speech (acoustic parameters) as close to natural speech as possible. The minimum generation error (MGE) criterion has been proposed to solve the second issue. It optimized the model parameters by minimizing the distortion between the generated speech parameters and the natural ones for the sentences in the training set. The MGE criterion has been applied not only to the clustered model training (Wu & Wang, 2006b) but also to the decision tree based model clustering of context-dependent models (Wu, Guo & Wang, 2006) and positive results have been achieved in improving the naturalness of synthetic speech. In (Wu, Guo & Wang, 2006), MGE was adopted to replace the ML criterion to select the optimal question at each tree node split. Since increasing the size of the decision tree always leads to the reduction of the generation error on the training set, MGE cannot be used directly as a stopping criterion in decision tree building. Thus, the size of the decision tree trained in (Wu, Guo & Wang, 2006) was tuned manually to compare the results with the MDL clustering that had almost equivalent numbers of leaf nodes.

On the other hand, cross-validation (CV) is a well-known technique to deal with the over-training and under-training problems without requiring extra development data. It estimates the accuracy of performance of a predictive model by partitioning the data set into complementary subsets and uses different subsets for training and validation (Bishop. 2006). In (Hashimoto, Zen, Nankaku, Masuko & Tokuda, 2009), a CV based method of setting hyper-parameters for HMM-based speech synthesis under the Bayesian criterion was proposed and positive results were reported.

In this paper, we integrate the minimum “cross” generation error criterion to optimize the size of the model clustering decision tree automatically for HMM-based speech synthesis. Different from (Wu, Guo & Wang, 2006), the ML criterion is still adopted to select the optimal question at each tree node split. A “cross” generation error is defined to calculate the sum of generation errors for all training sentences by cross-validation using the models clustered with a given decision tree. The size of the decision tree is optimized to minimize the cross generation error in two steps. First, an initial decision tree is obtained through model clustering with the MDL factor tuned with MCGE criterion. Then, the decision tree is finely modified by backing-off or splitting each leaf node iteratively to minimize the cross generation error. Objective and subjective evaluation results show that this proposed method outperforms the conventional MDL based HMM model clustering method significantly.

This paper is organized as follows: Section 2 describes the HMM-based speech synthesis method with conventional MDL clustering. In Section 3, the proposed MCGE based decision tree pruning method is introduced. Objective and subjective experimental results are discussed in Section 4. Finally, conclusions are given in Section 5.

2. HMM-based Parametric Speech Synthesis

2.1 The Framework of HMM-based Speech Synthesis

As shown in Figure 1, a typical HMM-based parametric speech synthesis system consists of two parts: the model training part and the speech synthesis part. In the model training part, spectrum, F0 and state duration are modeled simultaneously in a unified HMM framework. For each HMM state, the spectral features are modeled by a continuous probability distribution and F0 features are modeled using a multi-space probability distribution (MSD) (Tokuda, Masuko, Miyazaki & Kobayashi, 1999). In the synthesis step, speech parameters are generated from the trained models using maximum likelihood parameter generation (MLPG) algorithm (Tokuda, Yoshimura, Masuko, Kobayashi & Kitamura, 2000) and a parametric synthesizer is employed to reconstruct speech waveforms from the generated parameters.

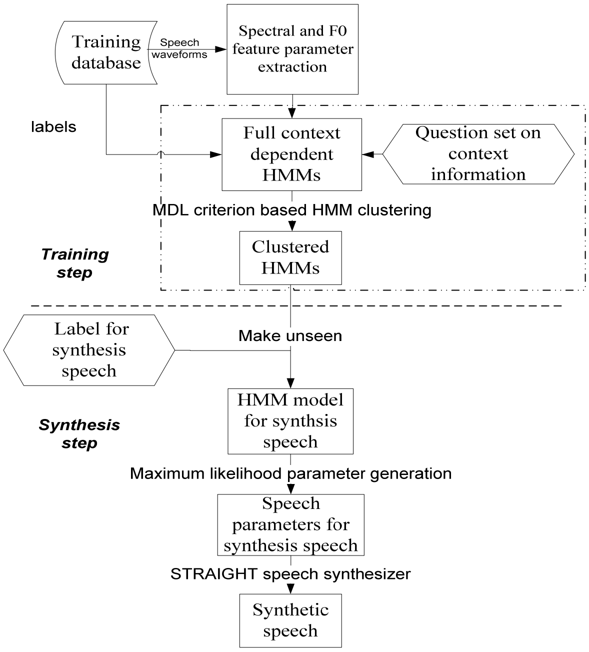

Training database
Speech waveforms
Spectral and F0 feature parameter extraction
labels
Full context dependent HMMs
Question set on context information
MDL criterion based HMM clustering
Training step
Clustered HMMs
Make unseen
Label for synthesis speech
Synthesis step
HMM model for synthsis speech
Maximum likelihood parameter generation
Speech parameters for synthesis speech
STRAIGHT speech synthesizer
Synthetic speech

Figure 1. Flowchart of a conventional HMM-based parametric speech synthesis system.

2.2 MDL-based Model Clustering

In the training stage, decision-tree-based model clustering is conducted after training for full context-dependent HMMs to avoid data-sparsity problems and to predict model parameters for the context features that do not occur in the training set. A question set containing language-dependent contextual questions is used. In the top-down decision tree building process, the ML criterion is commonly adopted to choose the optimal question and leaf node for splitting that lead to the greatest likelihood of growth. Further, the MDL principle is employed as a stopping criterion for decision tree pruning (Shinoda & Watanabe, 2000). The description length (DL) is defined as

$$I(\lambda) \equiv -\log P(\boldsymbol{o} \mid \lambda) + \frac{1}{2} D(\lambda) \log N + C \tag{1}$$

where λ denotes the clustered models; $\boldsymbol{o} = [\boldsymbol{o}_1^{\mathrm{T}}, \boldsymbol{o}_2^{\mathrm{T}}, \ldots, \boldsymbol{o}_N^{\mathrm{T}}]^{\mathrm{T}}$ is the training feature sequence, $(\cdot)^{\mathrm{T}}$ means the matrix transpose and N is the total frames of training data; $\log P(\boldsymbol{o} \mid \lambda)$ is the log likelihood function of λ on the training set; $D(\lambda)$ is the dimensionality of the model parameters; and C is a constant. The decision tree stops growth if the optimal leaf node splitting determined by the ML criterion can no longer reduce the DL.

If a single-Gaussian distribution with diagonal covariance matrix is used as the output probability distribution function (PDF) of each HMM state, Eq. (1) can be calculated as Equation (2) in (Shinoda & Watanabe, 2000)

$$I(\lambda) = \sum_{m=1}^{M} \frac{1}{2} \Gamma_m (E + E \log(2\pi) + \log|\boldsymbol{\Sigma}_m|) + EM \log N + C \tag{2}$$

where M is the leaf node number of the model clustering decision tree; Γ_m is the sum of state occupation probabilities for all frames in the training set belonging to the states that share the PDF of node m; E is the dimensionality of feature vectors; $\boldsymbol{\Sigma}_m$ is the covariance matrix of the Gaussian distribution function at node m.

Assume leaf node S with a contextual question is chosen among the M leaf nodes by ML criterion and further split into two child nodes SY and SN. Thus, the DL of the updated model λ' becomes

$$\begin{aligned} I(\lambda') = & \sum_{m=1, m \neq S}^{M} \frac{1}{2} \Gamma_m (E + E \log(2\pi) + \log|\boldsymbol{\Sigma}_m|) \\ & + \frac{1}{2} \Gamma_{SY} (E + E \log(2\pi) + \log|\boldsymbol{\Sigma}_{SY}|) \\ & + \frac{1}{2} \Gamma_{SN} (E + E \log(2\pi) + \log|\boldsymbol{\Sigma}_{SN}|) + E(M+1) \log N + C. \end{aligned} \tag{3}$$

The change of DL after the tree node splitting is

$$\Delta I = I(\lambda') - I(\lambda) = \frac{1}{2} \Gamma_{SY} \log|\boldsymbol{\Sigma}_{SY}| + \frac{1}{2} \Gamma_{SN} \log|\boldsymbol{\Sigma}_{SN}| - \frac{1}{2} \Gamma_S \log|\boldsymbol{\Sigma}_S| + E \log N. \tag{4}$$

The tree growth stops if $\Delta I > 0$. Thus, the stop condition of MDL-based decision tree building is

$$\frac{1}{2} \Gamma_S \log|\boldsymbol{\Sigma}_S| - \frac{1}{2} \Gamma_{SY} \log|\boldsymbol{\Sigma}_{SY}| - \frac{1}{2} \Gamma_{SN} \log|\boldsymbol{\Sigma}_{SN}| < E \log N. \tag{5}$$

The left side of Equation (5) presents the increase of log likelihood after the splitting. Therefore, the MDL criterion can be explained as introducing a threshold $E\log N$ into the ML-based decision tree construction. In practical system construction, an MDL factor $\alpha > 0$ is used to tune the threshold and control the size of the trained decision tree. Thus, Equation (5) can be rewritten as

$$\frac{1}{2}\Gamma_S \log|\Sigma_S| - \frac{1}{2}\Gamma_{SY} \log|\Sigma_{SY}| - \frac{1}{2}\Gamma_{SN} \log|\Sigma_{SN}| < \alpha E \log N. \tag{6}$$

Small α would lead to a large decision tree.

Besides MDL, the node size is also used as a complementary stop condition in practical system construction. It requires each leaf node to contain at least β samples otherwise the tree growth stops. Therefore, the pruning of the ML-trained model clustering decision tree is determined by a pair of parameters $\{\alpha,\beta\}$ with a default value of $\{1.0,15\}$ in our baseline system.

3. Minimum Cross Generation Error based Decision Tree Pruning

3.1 Cross Generation Error

In order to introduce MGE criterion into the pruning of model clustering decision tree, Cross Generation Error (CGE) is calculated on the training set by cross-validation. Assume the training database is composed of L sentences. To do cross-validation, we first divide the database into K subsets, $\{\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_K\}$ and

$$\boldsymbol{S}_k = \{\boldsymbol{C}_{k,1}, \boldsymbol{C}_{k,2}, ..., \boldsymbol{C}_{k,L_k}\}, k = 1, 2, ..., K \tag{7}$$

where $\boldsymbol{C}_{k,l} = [\boldsymbol{c}_{k,l,1}^{\mathrm{T}}, \boldsymbol{c}_{k,l,2}^{\mathrm{T}}, ..., \boldsymbol{c}_{k,l,T}^{\mathrm{T}}]^{\mathrm{T}}$ denotes the speech parameter sequence of the *l*-th sentence in the *k*-th subset, $\boldsymbol{c}_{k,l,t}$ is feature vector of the *t*-th frame in $\boldsymbol{C}_{k,l}$ and *T* is the frame number of $\boldsymbol{C}_{k,l}$; L_k is the number of sentences in subset *k* and $\sum_{k=1}^{K} L_k = L$. The phonetic balance needs to be considered when partitioning the database and the subsets should be divided as evenly as possible. When a model clustering decision tree *TR* is given, the “cross” generation error is calculated as

$$\mathcal{D}(TR) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{L_k}\sum_{l=1}^{L_k}\sum_{t=1}^{T_{k,l}} d(\boldsymbol{c}_{k,l,t}, \boldsymbol{c}'_{k,l,t}(\lambda_k(TR))) \tag{8}$$

where $\lambda_k(TR)$ represents the model estimated using the decision tree *TR* and the training subsets $\overline{\boldsymbol{S}}_k = \{\boldsymbol{S}_j\}_{j=1,...,K, j\neq k}$; $\boldsymbol{c}'_{k,l,t}(\lambda)$ denotes the generated parameter vector of frame *t* for the *l*-th sentence in subset *k* using model λ; $d(\boldsymbol{c}, \boldsymbol{c}')$ is an objective distortion function to calculate the generation error between the natural and generated speech parameters and a Euclidean distance measure is adopted here. The calculation process of the cross generation error is illustrated in Fig. 2.

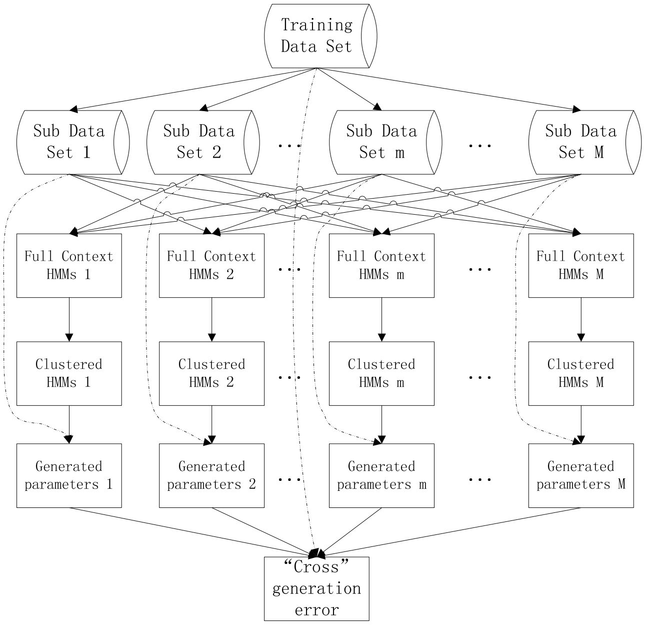


Figure 2. The calculation process of cross generation error.

3.2 Decision Tree Initialization

The pruning of the decision tree by CV and MGE is carried out in two steps. First we tune the MDL factor in Eq. (6) and the threshold in the node size stop condition discussed in Section 2.2 to generate an initial decision tree with a minimum cross generation error. Then the effect of each single tree leaf node on the cross generation error is inspected separately for further decision tree leaf backing-off or splitting. The decision tree initialization process is introduced in this section.

As shown in Equation (6), a small α would decrease the threshold in the stop condition of the MDL criterion and lead to a large decision tree. On the other hand, reducing the threshold β in the stop condition of the node size would also increase the size of the decision tree. A set of threshold parameter pairs $\{\alpha,\beta\}$ is designed in accordance with our speech synthesis system construction experience. For each pair of $\{\alpha,\beta\}$, a decision tree is trained via the method discussed in Section 2.2 and the cross generation error is calculated. We tune α first and keep β equal to its default value. When reducing α can no longer increase the size of the decision tree, we keep α constant and reduce β further. By such tuning, we are able to find a pair of $\{\alpha,\beta\}$ that leads to the smallest cross generation error. When the optimum pair of $\{\alpha,\beta\}$ is obtained, they are applied to conduct the model clustering using all of the training data and to generate the initial decision tree TR_0 for further optimization.

3.3 Cross Generation Error based Tree Pruning

Given an initial decision tree TR_0 by Section 3.2, the effect of every single leaf node on the cross generation error is inspected for further tree node back-off or splitting. Here, we define the cross generation error of tree node *m* as

$$\mathcal{D}_m(TR) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{L_k}\sum_{l=1}^{L_k}\sum_{t=1}^{T_{k,l}}\gamma_m(t)d(\boldsymbol{c}_{k,l,t},\boldsymbol{c}'_{k,l,t}(\lambda_k(TR))) \tag{9}$$

where $\gamma_m(t)$ denotes the state occupancy probability of frame *t* in the *l*-th sentence of subset *k* belonging to the node *m*. By comparing the sum of the cross generation error of each tree leaf node and its brother node with the cross generation error of their father node, it can decide whether we should back-off the leaf nodes to reduce the cross generation error or not. In the same way, we can decide whether the decision tree leaf should be split further. Backing-off or splitting continues for each decision tree leaf until no tree leaf can be backed-off or split. The optimization process for the decision tree backing-off and splitting is conducted iteratively and is described in detail as follows.

- ***Step 0.*** Given the divided training subsets $\{\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_K\}$ for cross-validation, the initial decision tree TR_0 is backed-off to get TR_1 to guarantee that each leaf node should contain at least one frame of sample from every $\overline{\boldsymbol{S}}_k$.
- ***Step 1.*** A group of clustered models $\{\lambda_k(TR_1)\}_{k=1,...,K}$ is estimated. Set $i=1$.
- ***Step 2.*** Back-off all the leaf nodes in TR_i to their father nodes by one level and attain TR_i' . Assume that leaf node *m* in TR_i' is the father node of node *ml* and *mr* in TR_i . If $\mathcal{D}_m(TR_i') < \mathcal{D}_{ml}(TR_i) + \mathcal{D}_{mr}(TR_i)$, we merge node *ml* and *mr* in TR_i into their father node. Otherwise, these two leaf nodes are reserved. This process is carried out for all leaf nodes in TR_i' and a new tree TR_{i+1} after necessary backing-off. Then set $i=i+1$. The flowchart of this backing-off process is shown in Fig. 3.
- ***Step 3***. Step 2 is repeated until the number of merged leaf nodes per one time back-off is smaller than a given threshold τ .
- ***Step 4.*** Splitting is conducted in a similar way after the backing-off process is finished.

Following these steps, decision tree TR_0 is finely tuned for every leaf, reducing the cross generation error on the training set.

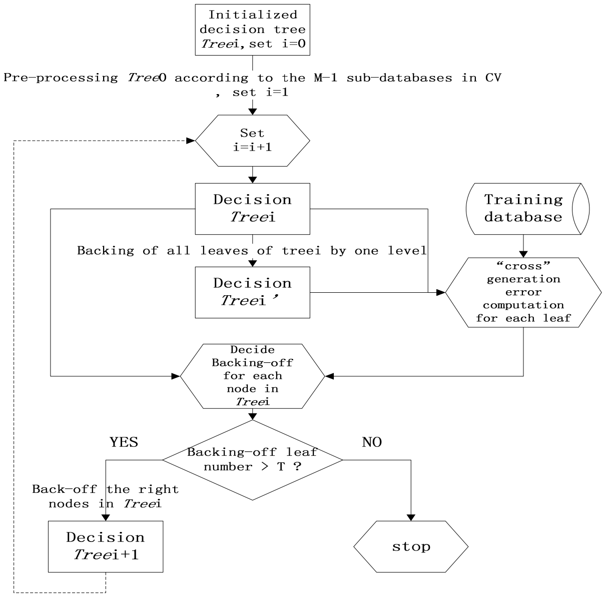

Initialized decision tree *Tree*i, set i=0
Pre-processing *Tree*0 according to the M-1 sub-databases in CV , set i=1
Set i=i+1
Decision *Tree*i
Training database
Backing of all leaves of treei by one level
Decision *Tree*i '
"cross" generation error computation for each leaf
Decide Backing-off for each node in *Tree*i
YES
NO
Backing-off leaf number > T ?
Back-off the right nodes in *Tree*i
Decision *Tree*i+1
stop

Figure 3. Flowchart for one decision tree back-off process.

4. Experiments

4.1 Experimental Conditions

In the experiment, we used a female phonetic balanced Mandarin database containing 1,000 sentences as the training database. The sample rate for the speech waves in the training database was 16kHz. 40 dimensional LSPs were extracted as the spectral features with 5ms frame shift. Five state context-dependent HMMs were used in the model training. Our experiments only focused on the decision-tree-based model clustering for spectral features. The context-dependent F0 and duration models were clustered in the conventional way.

A question set describing the contextual features for Mandarin Chinese was designed to conduct the decision tree splitting. The context features include:

- Left phone : phone before the current phone
- Current phone : the focused phone
- Right phone : phone after the current phone
- Left tone : tone of the syllable before the current syllable
- Current tone : the tone of the current syllable
- Right tone : tone of the syllable after the current syllable
- Part-of-speech : nature of the current word

- Relative positions of the current syllable, word, phrase, sentence, and sentence group
- Absolute positions from head and tail of the current syllable, word, phrase, sentence, and sentence group

4.2 Experiments on Decision Tree Initialization

4.2.1 Objective Evaluation

The training database was divided into ten subsets in our experiments. Following the method described in Section 3.2, a group of threshold parameter pairs $\{\alpha,\beta\}$ were designed as shown in Table 1. As the MDL factor α is the main factor that affects the size of the decision tree, we did not modify β until reducing α to where it could no longer enlarge the size of the decision tree. The System ID, the corresponding threshold parameter pairs $\{\alpha,\beta\}$, size of the decision tree, and the cross generation error calculated by LSP distortion introduced in Section 3.1 are shown Table 1.

Table 1. Scale of the decision tree and the objective LSF "cross" generation error for each system.

System ID	*Sys-A*	*Sys-B*	*Sys-C*	*Sys-D*	*Sys-E*	*Sys-F*	*Sys-G*	*Sys-H*	*Sys-I*
$\{\alpha,\beta\}$	{0.01,1}	{0.01,5}	{0.01,10}	{0.01,15}	{0.1,15}	{0.5,15}	{1,15}	{2,15}	{10,15}
Number of all leaf nodes	52882	36706	21211	14683	14654	8909	3946	1886	470
LSF distortion	0.02576	0.02498	0.02442	0.02421	0.02421	0.02428	0.02470	0.02553	0.02869

From Table 1, we can see that parameter set {0.01,15} (*Sys-D*) and {0.1,15} (*Sys-E*) lead to the smallest cross generation error. The baseline system is *Sys-G* with $\{\alpha,\beta\}$ in default settings.

4.2.2 Subjective Evaluation

A subjective listening test was also conducted for the above systems. As the trained decision trees of *Sys-D* and *Sys-E* were very close, *Sys-E* was omitted in the following subjective evaluation. Sixteen out-of-training-set test sentences were synthesized by the remaining eight systems. Five native Mandarin Chinese speakers were asked to give a score from 1 (very unnatural) to 5 (very natural) on the 128 synthetic sentences. The mean opinion scores (MOS) of all systems are shown in Fig. 4. From these results, we can see that the subjective scores match the objective cross generation error very well, where a smaller cross generation error corresponds to a higher MOS. *Sys-D* is the best system in the subjective evaluation and outperforms the baseline system (Sys-G). This proves the effectiveness of the proposed decision tree initialization method and the minimum cross generation error criterion. From

Figure 4 and Table 1, we also find that the LSF distortion of *Sys-A* and *Sys-B* is larger than *Sys-G*, but with a higher MOS score. This is reasonable because with a much smaller decision tree like in system *Sys-G*, the acoustic model would be too "average", making the synthesis speech "blurring". Nevertheless, large decision trees like *Sys-A* and *Sys-B* cause an over-training problem, where voice quality is not impacted much, but synthesized speech may not be stable.

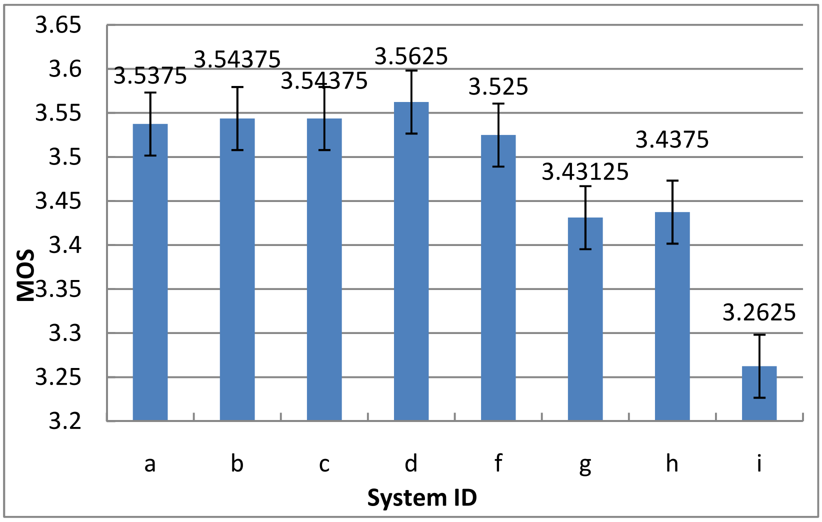


Figure 4. MOS of different systems for decision tree initialization.

4.3 Experiments on Decision Tree Pruning

4.3.1 Objective Evaluation

Using the threshold parameter pair {0.01,15} of *Sys-D*, the initial decision tree TR_0 was built by conducting MDL-based HMM clustering using this parameter set on the whole training database. Then further tree node backing-off and splitting introduced in Section 3.2 were conducted iteratively on the basis of TR_0. Here in the calculation of cross generation error, the same decision tree TR_0, other than the optimal $\{\alpha,\beta\}$, is utilized to conduct the model estimation of $\lambda_k(TR)$. The Euclidean LSP distance measure was used to compute the distortion between the generation and natural parameters. Figure 5 and Figure 6 describe the change in the cross generation error and the total number of the decision tree nodes in the iterative backing-off or splitting process. We can see that the cross generation error in Fig. 5 decreases consistently. Figure 6 shows that the backing-off was conducted for 9 iterations until no tree leaf could be backed-off and that node splitting was conducted for 2 iterations.

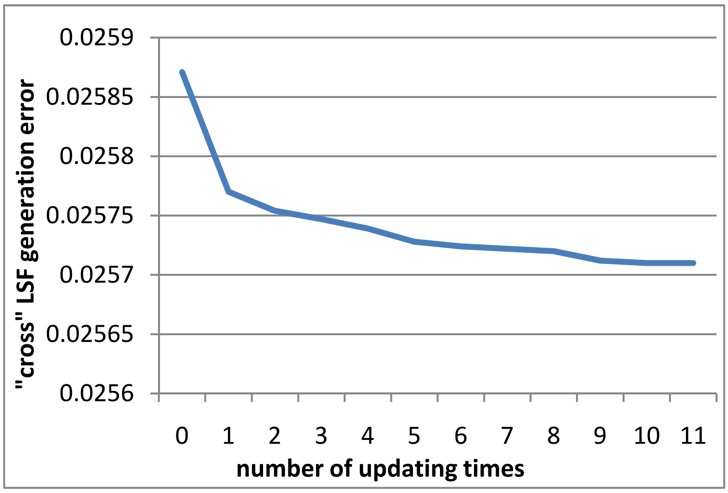


Figure 5. The "cross" generation error curve using Euclidean LSP distortion according to the decision tree pruning times. Decision tree backing-off is conducted 9 times until no leave can be combined. Then splitting for tree leaves is conducted for 2 times.

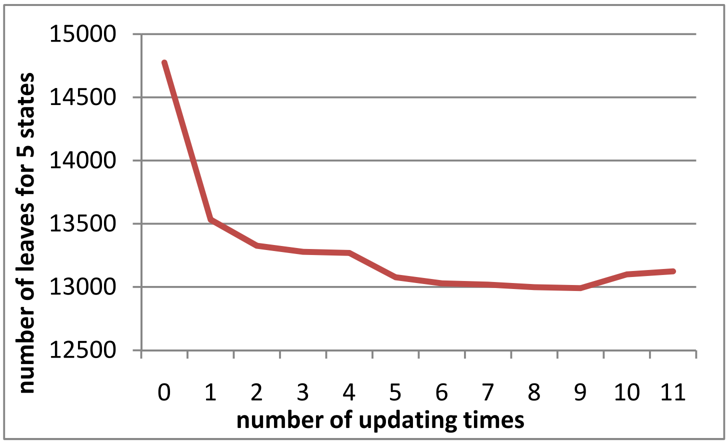


Figure 6. The scale of the decision tree according to the decision tree pruning times. Decision tree backing-off is conducted 9 times until no leave can be combined. Then splitting for tree leaves is conducted for 2 times.

Comparing Figure 5 and Table 1, one may find that the average "cross" generation error in the decision tree leaf backing-off and splitting process is larger than the average "cross" generation error in the MDL threshold parameter set optimizing process. This is normal because in the MDL threshold parameter optimization process, we employ the same MDL threshold parameter set for each $K-1$ sub-databases HMM clustering in the CV process. In the backing-off and splitting process, however, the same decision tree except for the MDL parameters is employed for HMM clustering in the CV. A different decision tree for different divisions in CV leads to a smaller “cross” generation error.

4.3.2 Subjective Evaluation

A subjective listening test was conducted for the three systems: the baseline system (*Sys-G*), the system with tuned $\{\alpha, \beta\}$ (*Sys-D*), and the system with further backing-off and splitting based on *Sys-D*. Sixteen sentences were synthesized by each of the systems and five native speakers were asked to choose the best sentence from the randomly ordered three sentences by three systems. The results are listed in Fig. 7, where the preference ratios for the three systems are 21.6%, 36.7% and 41.7% respectively.

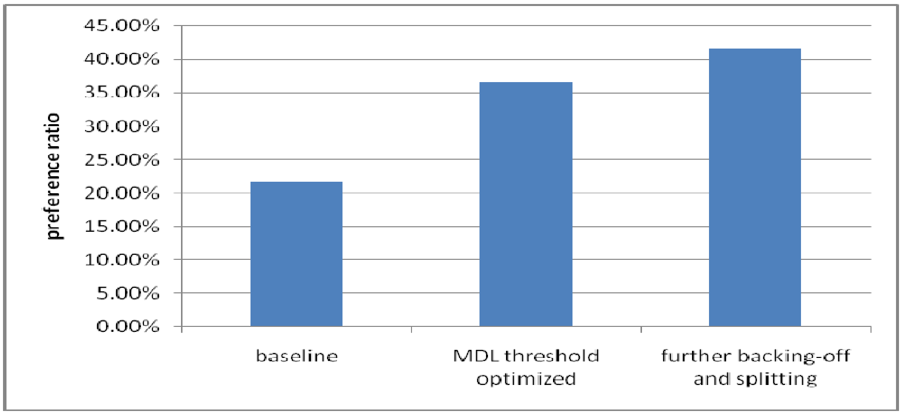


Figure 7. Preference ratio for the (1) baseline system, (2) MDL parameter optimized speech synthesis system and (3) further backing-off and splitting system.

From Figure 7, one can conclude that the MDL threshold parameter optimized speech synthesis system and further backing-off and splitting system both out-perform the baseline system. The proposed method for initialization of the decision tree and the further pruning method are both effective.

4.4 Discussion

The subjective MOS test and the objective LSP distortion prove the effectiveness of our two step decision tree pruning method. Compared with generating decision tree from the top or backing-off from the bottom, our two-steps decision tree pruning method, pruning the decision tree from the middle of the decision tree avoids many sub-optimums. If we start to prune from a huge decision tree which is split without any constraint using the method described in Section 3.3, we cannot guarantee that once the cross generation error by the father node is larger than the current tree leaves, the cross generation error by the grandfather level is also larger than the tree leaves. It could be smaller! Also pruning from the middle of the decision tree avoids a huge computational cost.

Theoretically, in order to get the decision tree that leads to the minimum cross generation error, one should use the minimum cross generation error criterion to choose the best question from the question set, and use the best question to conduct the splitting of every decision tree node. This means speech parameters for the synthesized speech should be generated and the cross generation error for the whole decision tree should be calculated for all the questions in the question set for each tree leaf. This will lead to an unacceptable computational cost. Another method of decision tree optimization is from the bottom to top. Using the ML criterion to conduct the decision tree generation with no stopping criterion, a huge decision tree is generated. In such a huge decision tree, there is almost only one sample for each tree leaf. Then the backing-off for each tree leaf to reduce the “cross” generation error is conducted. The problem, however, is that, backing-off the tree from the bottom does not always lead to the decision tree with the smallest “cross” generation error. It is quite possible that the backing-off process lead to some sub-optimal results. This is the case especially when there are only three tree leaves in the two level sub-tree. Nevertheless, informal experiments conducted by us revealed that, by conducting the decision tree leaf backing-off from the bottom of a huge decision tree as mentioned above, the out-of-training-set generation error of the optimized decision tree is even larger than the generation error by the decision tree initialized by only optimizing the MDL threshold parameters introduced in Section 3.2.

5. Conclusion

In this paper, we have proposed a minimum cross generation error criterion based decision tree pruning method for HMM-based parametric speech synthesis. Rather than generating the decision tree from the top or backing-off from the bottom, we optimize the decision tree from the middle. We first initialize the decision tree by tuning the MDL threshold parameter using the minimum “cross” generation error criterion over the whole decision tree. Then, by further backing-off or splitting tree leaves according to the cross generation error for every single leaf of the decision tree initialized in the first step, the optimal decision tree is obtained. In the decision tree pruning process, the cross generation error is calculated for every tree leaf using CV over the whole training database, and no extra development data set is needed.

In the experimental section, an objective cross generation error and subjective MOS score are both presented. The results show a smaller cross generation error leads to a higher MOS. Finally, subjective preference tests are conducted for the synthesized speech by comparing the baseline system, MDL threshold parameter optimized speech synthesis system and further backing-off and splitting system. The preference ratio indicates the effectiveness of our proposed method. The synthesized speech became more natural after the decision tree pruning process.

Acknowledgement

This work was partially supported by Hi-Tech Research and Development Program of China (Grant No.: 2006AA01Z137,2006AA010104) and National Natural Science Foundation of China (Grand No.: 60475015). The authors also thank the research division of iFlytek Co. Ltd., Hefei, China, for their help in corpus annotation.

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, U.S.A.

Black, A. W., Zen, H., & Tokuda, K. (2007). Statistical parametric speech synthesis. in *Proc. of ICASSP*, 4, 1229-1232.

Hashimoto, K., Zen, H., Nankaku, Y., Masuko, T., &Tokuda, K. (2009). A Bayesian approach to HMM-based speech synthesis. in *Proc. of ICASSP*, 4029-4032.

Hunt, A. J., & Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. in *Proc. of ICASSP*, 373-376.

Kataoka, S., Mizutani, N., Tokuda, K., & Kitamura, T. (2004). Decision-tree backing-off in HMM-based speech synthesis. In *Proc. of Interspeech*, 1205-1208.

Kawahara, H., Masuda-Katsuse, I., & Cheveigne, A. (1999). Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction: possible role of a repetitive structure in sounds. *Speech Commun*, 27 (3), 187-207.

Ling, Z. H., & Wang, R. (2007), HMM-based hierarchical unit selection combining Kullback-Leibler divergence with likelihood criterion. in *Proc. of ICASSP*, 1245-1248.

Rissanen, J. (1980). *Stochastic complexity in stochastic inquiry*.World Scientific Publishing Company.

Shinoda, K. & Watanabe, T. (2000). MDL-based context dependent subword modeling for speech recognition, *J. Acoust. Soc. Japan(E)*, 21(2), 79-86.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T. & Kitamura, T. (2000). Speech parameter generation algorithms for hmm-based speech synthesis. in *Proc. of ICASSP*, 3, 1315-1318.

Tokuda, K., Masuko, T., Miyazaki, N., & Kobayashi, T. (1999). Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. in *Proc. of ICASSP*, 229-232.

Wu, Y.-J., & Wang, R. (2006b). Minimum generation error training for HMM based speech synthesis. in *Proc. of ICASSP*, 89-92.

Wu, Y.-J., Guo, W., & Wang, R. (2006). Minimum generation error criterion for tree-based clustering of context dependent HMMs. in *Proc. of Interspeech*. 2046-2049.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., & Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. in *Proc. of Eurospeech*, 2347-2350.

The Association for Computational Linguistics and Chinese Language Processing

(new members are welcomed)

Aims：

1. To conduct research in computational linguistics.
2. To promote the utilization and development of computational linguistics.
3. To encourage research in and development of the field of Chinese computational linguistics both domestically and internationally.
4. To maintain contact with international groups who have similar goals and to cultivate academic exchange.

Activities：

1. Holding the Republic of China Computational Linguistics Conference (ROCLING) annually.
2. Facilitating and promoting academic research, seminars, training, discussions, comparative evaluations and other activities related to computational linguistics.
3. Collecting information and materials on recent developments in the field of computational linguistics, domestically and internationally.
4. Publishing pertinent journals, proceedings and newsletters.
5. Setting of the Chinese-language technical terminology and symbols related to computational linguistics.
6. Maintaining contact with international computational linguistics academic organizations.
7. Dealing with various other matters related to the development of computational linguistics.

To Register：

Please send application to:

The Association for Computational Linguistics and Chinese Language Processing
Institute of Information Science, Academia Sinica
128, Sec. 2, Academy Rd., Nankang, Taipei 11529, Taiwan, R.O.C.

payment： Credit cards(please fill in the order form), cheque, or money orders.

Annual Fees：

regular/overseas member： NT$ 1,000 (US$50.-)
group membership： NT$20,000 (US$1,000.-)
life member：ten times the annual fee for regular/ group/ overseas members

Contact：

Address： The Association for Computational Linguistics and Chinese Language Processing
Institute of Information Science, Academia Sinica
128, Sec. 2, Academy Rd., Nankang, Taipei 11529, Taiwan, R.O.C.

Tel.：886-2-2788-3799 ext. 1502 Fax：886-2-2788-1638

E-mail: aclclp@hp.iis.sinica.edu.tw Web Site: http://www.aclclp.org.tw

Please address all correspondence to Miss Qi Huang, or Miss Abby Ho

The Association for Computational Linguistics and Chinese Language Processing

Membership Application Form

Member ID#：__________________

Name：______________________________________ Date of Birth：_________________

Country of Residence：______________________ Province/State：______________________

Passport No.：_______________________________ Sex: __________________________

Education(highest degree obtained)：____

Work Experience：____

____

Present Occupation：____

Address：____

____

Email Add：____

Tel. No：__________________________________ Fax No：________________________________

Membership Category：□ Regular Member □ Life Member

Date： _____/_____/_____（Y-M-D）

Applicant's Signature：

Remarks： Please indicated clearly in which membership category you wish to register, according to the following scale of annual membership dues：

Regular Member	：	US$ 50.- （NT$ 1,000）
Life Member	：	US$500.-（NT$10,000）

Please feel free to make copies of this application for others to use.

Committee Assessment：

中華民國計算語言學學會

宗旨：

（一） 從事計算語言學之研究
（二） 推行計算語言學之應用與發展
（三） 促進國內外中文計算語言學之研究與發展
（四） 聯繫國際有關組織並推動學術交流

活動項目：

（一）定期舉辦中華民國計算語言學學術會議（Rocling）

（二）舉行有關計算語言學之學術研究講習、訓練、討論、觀摩等活動項目

（三）收集國內外有關計算語言學知識之圖書及最新發展之資料

（四）發行有關之學術刊物，論文集及通訊

（五）研定有關計算語言學專用名稱術語及符號

（六）與國際計算語言學學術機構聯繫交流

（七）其他有關計算語言發展事項

報名方式：

1. 入會申請書：請至本會網頁下載入會申請表，填妥後郵寄或E-mail至本會

2. 繳交會費：劃撥：帳號：19166251，戶名：中華民國計算語言學學會
信用卡：請至本會網頁下載信用卡付款單

年費：

終身會員：	10,000.-	（US$ 500.-）
個人會員：	1,000.-	（US$ 50.-）
學生會員：	500.-	（限國內學生）
團體會員：	20,000.-	（US$ 1,000.-）

連絡處：

地址：台北市115南港區研究院路二段128號 中研院資訊所(轉)
電話：(02) 2788-3799 ext.1502 傳真：(02) 2788-1638
E-mail：aclclp@hp.iis.sinica.edu.tw 網址: http://www.aclclp.org.tw
連絡人：黃琪 小姐、何婉如 小姐

中華民國計算語言學學會
個人會員入會申請書

會員類別	□終身 □個人 □學生	會員編號	（由本會填寫）		
姓　　名		性別		出生日期	年　月　日
				身分證號碼	
現　　職		學　　歷			
通訊地址	□□□				
戶籍地址	□□□				
電　　話		E-Mail			
申請人:　　　　　　　　（簽章） 中　華　民　國　　　年　　　月　　　日					

審查結果:

1. 年費：

終身會員：　10,000.-
個人會員：　1,000.-
學生會員：　500.-（限國內學生）
團體會員：　20,000.-

2. 連絡處：

地址：台北市南港區研究院路二段128號 中研院資訊所(轉)
電話：(02) 2788-3799　ext.1502　傳真：(02) 2788-1638
E-mail：aclclp@hp.iis.sinica.edu.tw　　網址: http://www.aclclp.org.tw
連絡人：黃琪 小姐、何婉如 小姐

3. 本表可自行影印

The Association for Computational Linguistics and Chinese Language Processing (ACLCLP)

PAYMENT FORM

Name :____(Please print) Date: ___________________

Please debit my credit card as follows: US$_______________________

□ VISA CARD □ MASTER CARD □ JCB CARD Issue Bank:_________________________

Card No.: __________- __________ - __________ - __________ Exp. Date:________________

3-digit code: ______________ (on the back card, inside the signature area, the last three digits)

CARD HOLDER SIGNATURE : ____________________________________

Tel.: ___________________________________ E-mail: ___________________________________

Add: ____

PAYMENT FOR

US$ _____ □ Computational Linguistics & Chinese Languages Processing (CLCLP)

Quantity Wanted: ______

US$ _____ □ Publications:___________________________________

US$ _____ □ Text Corpora: __________________________________

US$ _____ □ Speech Corpora:________________________________

US$ _____ □ Others: _______________________________________

US$ __________ □Life Member Fee □ New Member □Renew

US$ __________ = Total

Fax : 886-2-2788-1638 or Mail this form to :

ACLCLP
% Institute of Information Science, Academia Sinica
R502, 128, Sec.2, Academia Rd., Nankang, Taipei 115, Taiwan

E-mail: aclclp@hp.iis.sinica.edu.tw
Website: http://www.aclclp.org.tw

中華民國計算語言學學會
信用卡付款單

姓名: ________________________________(請以正楷書寫)　日期:：______________

卡別：□ VISA CARD □ MASTER CARD □ JCB CARD　發卡銀行：________________

卡號:__________-_________-_________- ________　有效日期：_________________

卡片後三碼：_____________________（卡片背面簽名欄上數字後三碼）

持卡人簽名：　__________________________________(簽名方式請與信用卡背面相同)

通訊地址：____

聯絡電話：_______________________　E-mail：_________________________________

備註：為順利取得信用卡授權，請提供與發卡銀行相同之聯絡資料。

付款內容及金額：

NT$__________□ 中文計算語言學期刊(IJCLCLP)

NT$__________□ 中研院詞庫小組技術報告

NT$__________□ 中文（新聞）語料庫

NT$__________□ 平衡語料庫

NT$__________□ 中文詞庫八萬目

NT$__________□ 中文句結構樹資料庫

NT$__________□ 平衡語料庫詞集及詞頻統計

NT$__________□ 中英雙語詞網

NT$__________□ 中英雙語知識庫

NT$__________□ 語音資料庫______________________________

NT$__________□ 會員年費　□續會　□新會員　□終身會員

NT$__________□ 其他: ___________________________________

NT$_______________=　合計

填妥後請傳真至 02-27881638 或郵寄至:
115台北市南港區研究院路2段128號中研院資訊所(轉)中華民國計算語言學學會 收
E-mail: aclclp@hp.iis.sinica.edu.tw
Website: http://www.aclclp.org.tw

Publications of the Association for Computational Linguistics and Chinese Language Processing

		Surface	AIR (US&EURP)	AIR (ASIA)	VOLUME	AMOUNT
1.	no.92-01, no. 92-04(合訂本) ICG 中的論旨角色與 A Conceptual Structure for Parsing Mandarin -- Its Frame and General Applications--	US$ 9	US$ 19	US$15	_____	_____
2.	no.92-02 V-N 複合名詞討論篇 & 92-03 V-R 複合動詞討論篇	12	21	17	_____	_____
3.	no.93-01 新聞語料庫字頻統計表	8	13	11	_____	_____
4.	no.93-02 新聞語料庫詞頻統計表	18	30	24	_____	_____
5.	no.93-03 新聞常用動詞詞頻與分類	10	15	13	_____	_____
6.	no.93-05 中文詞類分析	10	15	13	_____	_____
7.	no.93-06 現代漢語中的法相詞	5	10	8	_____	_____
8.	no.94-01 中文書面語頻率詞典（新聞語料詞頻統計）	18	30	24	_____	_____
9.	no.94-02 古漢語字頻表	11	16	14	_____	_____
10.	no.95-01 注音檢索現代漢語字頻表	8	13	10	_____	_____
11.	no.95-02/98-04 中央研究院平衡語料庫的內容與說明	3	8	6	_____	_____
12.	no.95-03 訊息爲本的格位語法與其剖析方法	3	8	6	_____	_____
13.	no.96-01 「搜」文解字－中文詞界研究與資訊用分詞標準	8	13	11	_____	_____
14.	no.97-01 古漢語詞頻表（甲）	19	31	25	_____	_____
15.	no.97-02 論語詞頻表	9	14	12	_____	_____
16.	no.98-01 詞頻詞典	18	30	26	_____	_____
17.	no.98-02 Accumulated Word Frequency in CKIP Corpus	15	25	21	_____	_____
18.	no.98-03 自然語言處理及計算語言學相關術語中英對譯表	4	9	7	_____	_____
19.	no.02-01 現代漢語口語對話語料庫標註系統說明	8	13	11	_____	_____
20.	Computational Linguistics & Chinese Languages Processing (One year) (Back issues of *IJCLCLP*: US$ 20 per copy)	---	100	100	_____	_____
21.	Readings in Chinese Language Processing	25	25	21	_____	_____
				TOTAL	_____	_____

10% member discount: ___________Total Due:__________

- **OVERSEAS USE ONLY**
- PAYMENT： □ Credit Card (Preferred)
  □ Money Order or Check payable to "The Association for Computation Linguistics and Chinese Language Processing " or “中華民國計算語言學學會”
- E-mail：aclclp@hp.iis.sinica.edu.tw

Name (please print): ____________________________ Signature: ____________________________

Fax: ____ E-mail: ______________________________

Address：____

中華民國計算語言學學會
相關出版品價格表及訂購單

編號	書目	會　員	非會員	冊數	金額
1.	no.92-01, no. 92-04 (合訂本)　ICG 中的論旨角色 與 A conceptual Structure for Parsing Mandarin--its Frame and General Applications--	NT$ 80	NT$ 100	_____	_____
2.	no.92-02, no. 92-03 (合訂本) V-N 複合名詞討論篇 與V-R 複合動詞討論篇	120	150	_____	_____
3.	no.93-01　新聞語料庫字頻統計表	120	130	_____	_____
4.	no.93-02　新聞語料庫詞頻統計表	360	400	_____	_____
5.	no.93-03　新聞常用動詞詞頻與分類	180	200	_____	_____
6.	no.93-05　中文詞類分析	185	205	_____	_____
7.	no.93-06　現代漢語中的法相詞	40	50	_____	_____
8.	no.94-01　中文書面語頻率詞典（新聞語料詞頻統計）	380	450	_____	_____
9.	no.94-02　古漢語字頻表	180	200	_____	_____
10.	no.95-01　注音檢索現代漢語字頻表	75	85	_____	_____
11.	no.95-02/98-04　中央研究院平衡語料庫的內容與說明	75	85	_____	_____
12.	no.95-03　訊息為本的格位語法與其剖析方法	75	80	_____	_____
13.	no.96-01　「搜」文解字－中文詞界研究與資訊用分詞標準	110	120	_____	_____
14.	no.97-01　古漢語詞頻表（甲）	400	450	_____	_____
15.	no.97-02　論語詞頻表	90	100	_____	_____
16	no.98-01　詞頻詞典	395	440	_____	_____
17.	no.98-02　Accumulated Word Frequency in CKIP Corpus	340	380	_____	_____
18.	no.98-03 自然語言處理及計算語言學相關術語中英對譯表	90	100	_____	_____
19.	no.02-01 現代漢語口語對話語料庫標註系統說明	75	85	_____	_____
20	論文集 COLING 2002 紙本	100	200	_____	_____
21.	論文集 COLING 2002 光碟片	300	400	_____	_____
22.	論文集 COLING 2002 Workshop 光碟片	300	400	_____	_____
23.	論文集 ISCSLP 2002 光碟片	300	400	_____	_____
24.	交談系統暨語境分析研討會講義 （中華民國計算語言學學會1997第四季學術活動）	130	150	_____	_____
25.	中文計算語言學期刊（一年四期）　年份：______ （過期期刊每本售價500元）	---	2,500	_____	_____
26.	Readings of Chinese Language Processing	675	675	_____	_____
27.	剖析策略與機器翻譯 1990	150	165	_____	_____
			合　計	_____	_____

※　此價格表僅限國內 (台灣地區) 使用

劃撥帳戶：中華民國計算語言學學會　　劃撥帳號：19166251

聯絡電話：(02) 2788-3799 轉1502

聯絡人：　黃琪 小姐、何婉如 小姐　　E-mail:aclclp@hp.iis.sinica.edu.tw

訂購者：　________________________　收據抬頭：______________________________

地　　址：____

電　　話：__________________________　E-mail: ____________________________

Information for Authors

International Journal of Computational Linguistics and Chinese Language Processing (IJCLCLP) invites submission of original research papers in the area of computational linguistics and speech/text processing of natural language. All papers must be written in English or Chinese. Manuscripts submitted must be previously unpublished and cannot be under consideration elsewhere. Submissions should report significant new research results in computational linguistics, speech and language processing or new system implementation involving significant theoretical and/or technological innovation. The submitted papers are divided into the categories of regular papers, short paper, and survey papers. Regular papers are expected to explore a research topic in full details. Short papers can focus on a smaller research issue. And survey papers should cover emerging research trends and have a tutorial or review nature of sufficiently large interest to the Journal audience. There is no strict length limitation on the regular and survey papers. But it is suggested that the manuscript should not exceed 40 double-spaced A4 pages. In contrast, short papers are restricted to no more than 20 double-spaced A4 pages. All contributions will be anonymously reviewed by at least two reviewers.

Copyright： It is the author's responsibility to obtain written permission from both author and publisher to reproduce material which has appeared in another publication. Copies of this permission must also be enclosed with the manuscript. It is the policy of the CLCLP society to own the copyright to all its publications in order to facilitate the appropriate reuse and sharing of their academic content. A signed copy of the IJCLCLP copyright form, which transfers copyright from the authors (or their employers, if they hold the copyright) to the CLCLP society, will be required before the manuscript can be accepted for publication. The papers published by IJCLCLP will be also accessed online via the IJCLCLP official website and the contracted electronic database services.

Style for Manuscripts: The paper should conform to the following instructions.

1. Typescript: Manuscript should be typed double-spaced on standard A4 (or letter-size) white paper using size of 11 points or larger.

2. Title and Author: The first page of the manuscript should consist of the title, the authors' names and institutional affiliations, the abstract, and the corresponding author's address, telephone and fax numbers, and e-mail address. The title of the paper should use normal capitalization. Capitalize only the first words and such other words as the orthography of the language requires beginning with a capital letter. The author's name should appear below the title.

3. Abstracts and keywords: An informative abstract of not more than 250 words, together with 4 to 6 keywords is required. The abstract should not only indicate the scope of the paper but should also summarize the author's conclusions.

4. Headings: Headings for sections should be numbered in Arabic numerals (i.e. 1.,2....) and start form the left-hand margin. Headings for subsections should also be numbered in Arabic numerals (i.e. 1.1. 1.2...).

5. Footnotes: The footnote reference number should be kept to a minimum and indicated in the text with superscript numbers. Footnotes may appear at the end of manuscript

6. Equations and Mathematical Formulas: All equations and mathematical formulas should be typewritten or written clearly in ink. Equations should be numbered serially on the right-hand side by Arabic numerals in parentheses.

7. References: All the citations and references should follow the APA format. The basic form for a reference looks like

Authora, A. A., Authorb, B. B., & Authorc, C. C. (Year). Title of article. *Title of Periodical, volume number*(issue number), pages.

Here shows an example.

Scruton, R. (1996). The eclipse of listening. *The New Criterion, 15*(30), 5-13.

The basic form for a citation looks like (Authora, Authorb, and Authorc, Year). Here shows an example. (Scruton, 1996).

Please visit the following websites for details.

(1) APA Formatting and Style Guide (http://owl.english.purdue.edu/owl/resource/560/01/)

(2) APA Stytle (http://www.apastyle.org/)

No page charges are levied on authors or their institutions.

Final Manuscripts Submission: If a manuscript is accepted for publication, the author will be asked to supply final manuscript in MS Word or PDF files to clp@hp.iis.sinica.edu.tw

Online Submission: http://www.aclclp.org.tw/journal/submit.php

Please visit the IJCLCLP Web page at http://www.aclclp.org.tw/journal/index.php

International Journal of
Computational Linguistics & Chinese Language Processing
vol.15 no.1 March 2010

Contents

Papers

ISSN: 1027-376X
The Association for Computational Linguistics and Chinese Language Processing