

Knowledge extraction through etymological networks: Synonym discovery in Sino-Korean words

Pablo E.* , Kyomin Jung†
 pablo@snu.ac.kr* , kjung@snu.ac.kr†
 Seoul National University, Korea

Abstract—Extracting knowledge from a text is a very active area of research. Techniques such as word embedding and LSA have brought great breakthroughs and have been used in applications such as automatic translation. We propose a novel approach to extract knowledge from text that relies on a graph to express the complex etymological structures formed by the historical roots of words.

Our approach is specially fit for the study of Sino-Korean vocabulary, where the etymological roots of words are clearly shown in their writing. We use our approach to build a bipartite graph based on the Chinese etymological roots of Sino-Korean words, and then use the network structure to extract features describing pairs of nodes. We used these features in a classification scheme to discover pairs of nodes that represent synonym characters. Our model is simpler than previous work on synonym discovery with Chinese characters, and obtains good results. The code and data for our work are made openly available.

Index Terms—Graph mining, edge classification, Chinese characters, Korean language.

I. INTRODUCTION

In the last few years, it has been found that many real-world systems can be modeled through networks (e.g. the internet, conflicts between countries). By modeling complex relationships with networks, researchers have been able to extract valuable insights from their intricate structures. Indeed, through the techniques of graph theory, many remarkable observations have been made in fields such as economics and social science [1], food pairings [2], and even organized crime [3].

In this work, we attempt to model the vocabulary of a human language as a network. We do this by studying the historical roots of words. It is known that words in human languages are often formed by mixing and matching smaller *meaning units*, also known as etymological roots. For example, the word *autonomous* is integrated by its roots *autos*, which means *self*; and *nomos*, which means *law*: therefore, through study of the etymological roots of the word *autonomous*, we can derive that its meaning is *self-governing*. Our approach, in simple terms, consists on using this technique at a much larger scale.

Determining the etymological roots of words in English normally requires a lot of manual work to trace words back to their ancestors. This is perhaps the reason that not much work has been done in the areas of data mining and knowledge management that is based on etymology. On the other hand, etymology discovery is not a so significant problem with Sino-Korean words because unlike phonetic alphabets, which carry only sound information, Chinese characters carry information

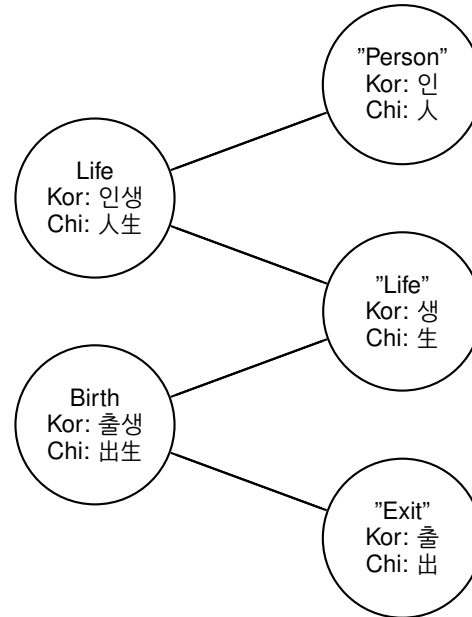


Fig. 1. Subset of bipartite graph. On the left are the ‘words’ that are formed by mixing ‘roots’, which can be seen on the right. Most words are formed by using two etymological roots, but there are also words formed by more than two roots.

about their meaning. This means that upon reading the Chinese spelling of a Sino-Korean word (formed by one or more individual characters) it is possible to know the *meaning units* that integrate it.

These relationships between *meaning units* and words in a language can be used to define a natural bipartite network, the structure of which can be seen in Fig.1. A graph of this kind contains information about the structure of vocabulary and its relationships. As part of our work, we set out to discover whether the techniques from graph theory can be used on an etymological graph like this to bring some insights to the field of computational linguistics.

There have been some studies of Chinese etymology through the lens of network science. Previous research has mostly paid attention to how radicals integrate to form more complex characters [4], but only one study has focused on words formed by one or more characters [5]. In said study the authors analyze the characteristics of the network, and propose a scheme to generate similar random graphs. To the best of our knowledge, no other studies have tried to extract knowledge from the structure of an Etymological network.

A. Our work and contributions

In this paper we define a framework for creating etymological networks of any language, and use it to build an etymological graph of Sino-Korean words. We then use this graph to build a new graph of Chinese characters, where two characters are connected if they are able to form a word together. After this, we used the second graph to discover pairs of Chinese characters that are synonyms.

To discover pairs of synonyms, we defined a classification scheme where we use supervised learning to classify pairs of characters as synonyms or non-synonyms depending on features of the graph, the nodes that represent them, and the neighborhood of these nodes. The classifier that we used is a Random Forest. Our classifier was able to reliably find new pairs of synonyms with high precision and acceptable recall. We succeeded in using an etymological graph to obtain semantic information. The code for this paper, as well as the datasets and instructions on how to replicate this work are openly available [6].

However, an important feature of an etymological graph-based framework is that it can support several different techniques to extract semantic knowledge, and supervised learning-based synonym discovery is only one of them. We are very interested on seeing this framework applied in other ways. Other techniques that we believe can be utilized are LSA [7], clustering, and similarity matrix [8].

II. METHOD

A. Building the etymological graph

An etymological graph, as we define it, is a bipartite network where one set of nodes represents the roots of the words in a language, and the other set of nodes represents the words themselves. We built the graph using etymological information that we obtained by crawling the dictionary of a Korean online portal. We started crawling from a single Chinese character (e.g. "Large" Kor: 대 Chi: 大), and obtaining all the words that use that character. Afterwards, we obtained all the characters that those words used, and with those, we obtained the words that use them. Upon retrieving a new word or character, they would be added to the graph, and connected to their related nodes.

By following that strategy, we built a graph where two nodes would be connected if and only if one node represents an etymological root of the word represented by the other node. For example, given a vocabulary with the words Life (Kor: 인생 Chi: A生), and Birth (Kor: 출생 Chi: 2生), our graph would have one node to represent each one of those words, and three more nodes that represent their etymological roots, which in this case are the individual Chinese characters A, 生 and 2. This example graph is shown in Fig.1. By building a graph in this manner, we are able to represent the complex structure of the vocabulary of a language. This is because every edge in the graph represents a strong relationship between a word and one of its historical roots.

Because the graph was built by crawling, it is a fully connected graph (i.e. there exists a path that connects every node with any other node in the graph). We obtained a graph

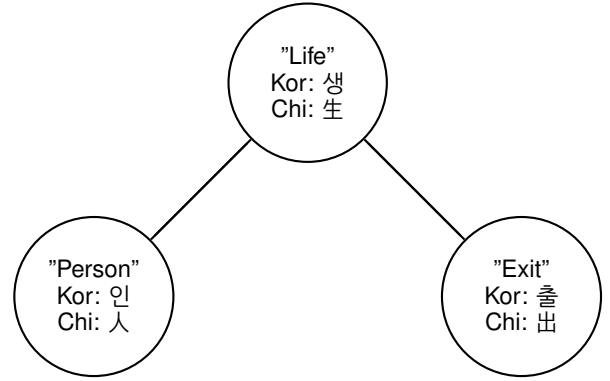


Fig. 2. Chinese character unipartite projection of the bipartite graph shown before. The character '生' participates in both words, and is therefore connected to all other characters. The characters '人' and '出' do not form a word together, and are therefore not connected.

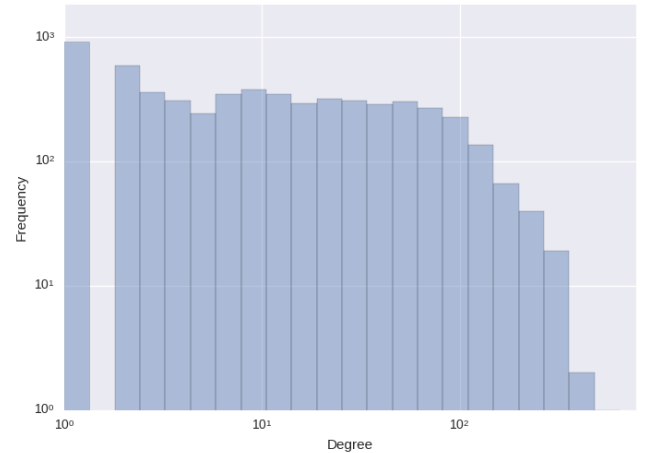


Fig. 3. Degree distribution of the unipartite root-projection of the graph in log-log scale. The degree of a node in the unipartite root-projection is a measure of how many words a character forms.

that consists of 5 955 Chinese characters (roots), and 136 045 Sino-Korean words. To be sure that our crawling strategy was thorough, we made sure that these 5 955 characters included all 1 800 basic characters covered in the school curriculum in South Korea, which they did.

B. Using graph features in supervised training

From our initial bipartite network, we attempted to discover pairs of Chinese characters that are synonyms while using only information derived from the graph structure. To do this, we would only work with root information, and thus we obtained the root-unipartite projection of our initial bipartite graph (i.e. the projection that contains only nodes representing etymological roots) [14]. This gave us a network integrated of nodes representing only Chinese characters. In this new network, two nodes share a link only if the characters that they represent can form a word together. This is similar to the graph built in [5]. This transformation can be visualized with the graph in Fig.2, which is the root-unipartite projection of the

TABLE I
FEATURES CALCULATED FOR EACH PAIR OF NODES

Feature name	Formula
Reciprocity ¹	$A \in \text{neighbors}(B)$
Self-neighbor degree ratio	$\left \frac{\text{degree}(A)}{\text{average_degree}(\text{neighbors}(A))} - \frac{\text{degree}(B)}{\text{average_degree}(\text{neighbors}(B))} \right $
Normalized count of two-step walks from A to B	$\frac{\text{two_step_walks}(A,B)}{\min(\text{degree}(A), \text{degree}(B))}$
Absolute degree difference	$ \text{degree}(A) - \text{degree}(B) $
Sum of their degrees	$\text{degree}(A) + \text{degree}(B)$
Normalized degree ratio	$\frac{\min(\text{degree}(A), \text{degree}(B))}{\max(\text{degree}(A), \text{degree}(B))}$
Sum of clustering coefficients [9]	$c_A + c_B$
Absolute difference of clustering coefficients	$ c_A - c_B $
Product of clustering coefficients	$c_A \cdot c_B$
Dispersion between nodes [10]	$\text{disp}(A, B)$
Sum of their betweenness centrality [11]	$\text{betweenness}(A) + \text{betweenness}(B)$
Absolute difference of their betweenness centrality	$ \text{betweenness}(A) - \text{betweenness}(B) $
Sum of their closeness centrality [12]	$\text{closeness}(A) + \text{closeness}(B)$
Absolute difference of their closeness centrality	$ \text{closeness}(A) - \text{closeness}(B) $
Sum of their PageRank scores [13]	$PR(A) + PR(B)$
Absolute difference of their PageRank scores	$ PR(A) - PR(B) $
Distance between nodes	$\text{length}(\text{path}(A, B))$
Belong to same 6-clique	$\exists C \in \mathcal{C}_6 \mid A \in C \wedge B \in C$
Belong to the same 7-clique	$\exists C \in \mathcal{C}_7 \mid A \in C \wedge B \in C$

TABLE II
RESULT FROM THE SYNONYM CLASSIFICATION SCHEME

Problem	Pairs of nodes to classify	Training synonyms	Probability cutoff	Classified as synonyms	Discovered synonyms	Precision	Recall
Classification of free pairs	17 653 143	2 889	0.5	2 667	2 160	0.81	0.68
Link classification	74 892	1 618	0.8	1 565	1 315	0.84	0.47

graph shown in Fig.1. Fig.4 shows a subgraph of the resulting unipartite projection from our data.

Then we attempted to extract synonym knowledge from the graph structure. To do this we applied supervised learning by classifying pairs of nodes as synonyms or non-synonyms. To obtain synonym data, we collected it from the dictionary of a Korean online portal and obtained approximately 5 000 pairs of Chinese characters that are synonyms. For non-synonym pairs, we obtained 5 000 random samples over all possible pairs of nodes, a step which is likely to have added some acceptable output noise to the classifier.

To apply a scheme that classifies pairs of nodes as synonym or non-synonym, we computed 19 features for each pair of nodes in the unipartite projection of the network. These features are detailed in Table I. The features were selected to describe the immediate neighborhood between pairs of nodes (e.g. reciprocity, shortest distance, number of two-step walks, dispersion, same 6/7 clique), and to describe characteristics of the nodes themselves (e.g. degree, PageRank, Centrality and Betweenness sums/differences). To manage the graph and calculate these features we used the NetworkX python library [15].

Because we work with almost 6000 characters, and we want to classify pairs of them as synonyms, the size of our search space is on the neighborhood of 18 000 000 pairs of Chinese characters, and the set of synonyms is a very small fraction of this set. Given these conditions, and to improve our classification scheme, we constructed two models: One for pairs of nodes that shared an edge (i.e. that formed a word

together), and another for pairs of nodes that did not share an edge.

This is useful because, unlike in the English language, Chinese characters *often* pair with synonym characters to express the same idea (e.g. the word $\dot{\cup}$ ('sameness'), where $\dot{\cup}$ means 'one', and $\dot{\cup}$ means 'same'). By constructing two models, we allow each model to focus on its specific task, and improve predictive power.

We explored several classification techniques, but obtained best results when using boosted Random Forests ($N_{trees} = 10$, $size_{tree} = \sqrt{N}$) using the Python scikit-learn package [16].

III. RESULTS

A. The edge classification problem

Because we were only considering pairs of nodes that share an edge, this section can be thought of as an edge classification problem; where we want to determine if an edge is linking two synonyms, or not. This search space is of manageable size - i.e. the Chinese character uniprojection has a total of 74 893 edges. Because the space of edges is of manageable size, the recall (i.e. the percentage of *known* synonyms that were classified as such) and the precision (i.e. the percentage of pairs classified as synonyms that are actual synonyms) of the classification scheme were very high even with a standard probability cutoff ($P(\text{syn}) = 0.5$), as can be seen in Table II.

¹Reciprocity was used to separate edge classification from node pair classification, thus it was not relevant for the Random Forest classifier.

B. The node-pair classification problem

This can be thought of as a link prediction problem, but only of synonym-typed links. The search space is much larger, since for 5955 Chinese characters there can be a total of 17 728 035 different combinations that must be classified, out of which only very few yield real synonyms. Because of this, we had to adjust the classification scheme to yield only pairs of nodes that are very likely to be synonyms according to our random forest ($P(\text{syn}) \approx 0.8$); otherwise the precision of the classification scheme would be too low. The results can also be seen in Table II.

C. Outcome

The model was able to discover a total of 3475 new pairs of synonyms from both schemes. It erroneously classified 757 pairs of characters as synonyms. Also, although the model was able to discover a significant number of real new pairs of synonyms, the recall of the known synonyms (used for testing) was less than desirable. We believe that this is due to 1) the features that we chose being deficient or insufficient, and therefore not able to properly describe the neighborhood of a pair of nodes, or 2) the *synonym space* being too large to be described by our available known data. Perhaps, instead of depending on the features that we hand-code and a classification setup, tackling the problem with unsupervised approach or a purely graph-theoretic one, where the network structure is more intimately described in the learning process would help improve the recall of a classification scheme like this one.

Nonetheless, our approach is robust and clear. Other attempts to find synonyms in Chinese vocabulary have been explored in [17], and [18]. Specifically, [18] uses a co-occurrence matrix approach, which is similar to our graph-based approach. The network on [18] is built using co-occurrence in a lexical context, as opposed to co-occurrence in an etymological context. By using etymology we can bypass noise coming from a free corpus.

Another interesting result from the experiments is that the model identified some antonyms as being synonyms (e.g. 硬 'hard' and 柔 'soft'; 始 'start' and 終 'end'; 母 'mother' and 父 'father'). This might stem from the fact that antonym characters are often interchangeable in words (e.g. 'exit' and 'entrance' in 出門 'exit door' and 入門 'entrance door'), so their features are almost the same, and they exist very close in the feature space.

IV. CONCLUSION AND FUTURE WORK

On this work, an etymological graph has been successfully used for knowledge extraction, specifically in the form of synonym discovery. We believe that etymological graphs can be used in more diverse applications. We are particularly interested in seeing these techniques being applied to different languages.

One important challenge for this approach to be used in languages with phonetic alphabets is Etymology extraction. Etymological dictionaries are able to significantly bridge this gap. Nonetheless, if we consider the fact that globalization has

allowed for words to continue to be borrowed and coined at a higher pace, the task becomes more complicated; making it more difficult to construct simple root-word graphs.

However, we think that our approach, paired with tools of text mining, and etymological dictionaries can be used to refine the technique and mine knowledge from languages that use both phonetical alphabets and logographic writing systems. Research has very rarely approached the problem of semantic knowledge through etymological networks, with only a few exceptions [19] [20].

Future work will focus on improving synonym discovery, and exploring other techniques to derive semantic knowledge from the etymological graph of Sino-Korean words. An interesting possibility is to use Latent Semantic Analysis [7] over the matrix of the bipartite graph. We believe that topic modeling has strong parallels with synonym discovery and word embedding in general. We are very interested in seeing new techniques being applied with an etymological graph-based approach.

REFERENCES

- [1] M. O. Jackson *et al.*, *Social and economic networks*, vol. 3. Princeton university press Princeton, 2008.
- [2] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, "Flavor network and the principles of food pairing," *Scientific reports*, vol. 1, 2011.
- [3] M. Kenney, "The architecture of drug trafficking: network forms of organisation in the colombian cocaine trade," *Global crime*, vol. 8, no. 3, pp. 233–259, 2007.
- [4] J. Li and J. Zhou, "Chinese character structure analysis based on complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 380, pp. 629–638, 2007.
- [5] K. Yamamoto and Y. Yamazaki, "A network of two-chinese-character compound words in the japanese language," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 12, pp. 2555–2560, 2009.
- [6] E. Pablo, "hanja-graph - software for crawling and analysis of sino-korean etymological networks." <https://github.com/pabloem/hanja-graph>.
- [7] S. T. Dumais, "Latent semantic analysis," *Annual review of information science and technology*, vol. 38, no. 1, pp. 188–230, 2004.
- [8] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren, "A measure of similarity between graph vertices: Applications to synonym extraction and web searching," *SIAM review*, vol. 46, no. 4, pp. 647–666, 2004.
- [9] J. Saramäki, M. Kivela, J.-P. Onnela, K. Kaski, and J. Kertesz, "Generalizations of the clustering coefficient to weighted complex networks," *Physical Review E*, vol. 75, no. 2, p. 027105, 2007.
- [10] L. Backstrom and J. Kleinberg, "Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook," pp. 831–841, 2014.
- [11] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.
- [12] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.
- [13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web," 1999.
- [14] M. E. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 1, pp. 2566–2572, 2002.
- [15] D. A. Schult and P. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, vol. 2008, pp. 11–16, 2008.
- [16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [17] L. Yong, Z. Chengzhi, and H. Hanqing, "Using multiple hybrid strategies to extract chinese synonyms from encyclopedia resources [j]," *Journal of Library Science in China*, vol. 1, p. 010, 2010.

