

SHANGHAI JIAO TONG UNIVERSITY

**学士学位论文**

BACHELOR’S THESIS



论文题目：

学生姓名:

学生学号:

专 业:

指导教师:

学院(系):

**Contents**

[Chapter One Introduction 3](#_Toc483670680)

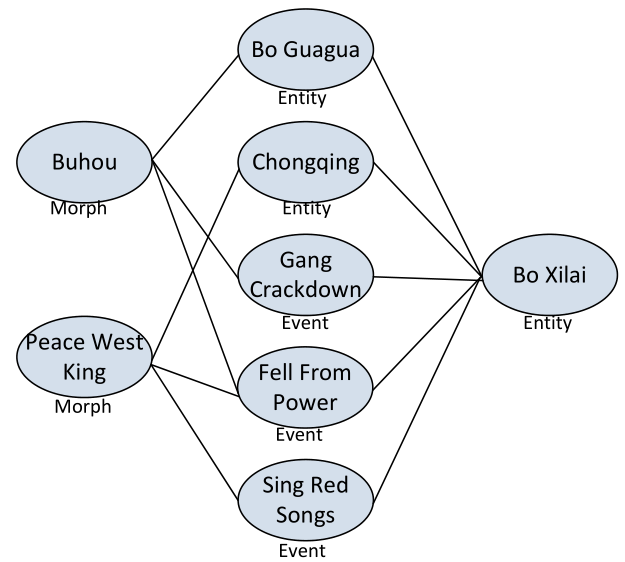
[1.1 Section 3](#_Toc483670681)

Chapter One Introduction

The 21st century is an information explosion era. The word ‘Big Data’ is presented to describe the phenomena that numerous amount of data is being generated in the current daily life. Information is exchanged in a rapid speed through Internet. Hotspot no longer springs just one month a time, it springs every day. Moreover, with the wide spreading of smart phone, it has become much more easily for people to fetch the latest information. People currently are used to exploring interesting staff or other information on online social platforms/community, like Sina Weibo, Wechat Group, Zhihu. New terms are created and spread widely since some event that attracts a lot of people happens. These kinds of new terms have a strong relation to the entity they refer, like ‘平西王’ refers to ‘薄熙来’ due to the similarity between these two entities. We call this kind of new term as ‘Morph’, while the one that it refers is called ‘Target Entity’. But due to the aging effect, people who are not familiar with the background of the creation of the morph may feel confused about the origin of the new term. Only when the relevant event is still the hot spot can people easily find the reason of the origination, because numerous We-Media write blogs to analyze and explain the event, which will introduce the meaning of the morph. So when the period of the hot spot passes, it is hard to understand the originating relation between the morph and the target entity. Moreover, due to the blockade of the Internet, some web documents are not accessible to most of people, which aggravate the blindness to understanding the originating relation. Thus it is meaningful to find a way, which can explain such kind of latent relation. This is the purpose of my project.

* 1. Previous Work

Before I presented this project, I’ve read three papers about morph, which inspired me to do this project. Following the temporal order, they are ‘Resolving Entity Morphs in Censored Data’(Huang et al., 2013), ‘Be Appropriate and Funny: Automatic Entity Morph Encoding’ (Zhang et al. 2014), ‘Context-aware Entity Morph Decoding’(Zhang et al. 2015). The first and third paper is to find out morphs from corpus, namely to detect the entity and further detect whether it is a morph or not. The second paper is to generate morphs from some entities. The first paper uses a graph to compute the similarity between a target morph and the base entity, as the Figure X shows. Figure X is the ‘Figure 4: Example of Morph-Related Heterogeneous Information Network’ in (Huang et al., 2013)’s paper.



It calculates the similarity based on the graph, along with other prior knowledge, such as the social features. This graph gives me a lot of inspiration. If machine can learn the strong relation between a morph and its target entity, human beings can also learn from it. But this relation network in Huang’s paper is used to candidate the best morph. It’s a comparison between the numbers. So the absolute number of the score may not be able to reflect the originating relation between the morph and its target entity. The third paper pays more attention on the morph verification and morph candidate ranking. The second paper introduces 7 ways to generate a man-made morph, from phonetic view, spelling view, nickname view, translation and transliteration view, semantic view, historical figure view, and characteristics view. These views show clearly how a morph can relate to an entity.

* 1. Similar Problems

The problem, explaining the originating relation between a morph and its target entity, is very similar to reason the relation between two entities from the Knowledge Graph. This word is presented by Google, whose purpose is to enhance its search engine. This graph is useful in semantic searching. The query like ‘Who is Thomas Jefferson?’ is processed with the help of Knowledge Graph. The nodes within this graph represents the real entities that exist in the real world, the edges within this graph represents the relations between the entities. You may get the result like ‘Born: April 13, 1743’ from this query, inside which ‘Born’ is the relation edge and ‘April 13, 1743’ is the entity node. As to my project, the query may become ‘Why 平西王 refers to 薄熙来?’. It may tell you that they are both related to ‘重庆’, ‘政府’. So it is very similar. But the difference is, to explain the originating relation between the morph and its target entity is a subset of the general Knowledge Graph query, also it needs a more complicated explanation other than just tell you a simple list of relation-entity tuples.

You may hear another word ‘Linked Data’. This term is created by Tim Berners-Lee (the director of W3C), which is a publishing method, aiming to link data to help semantic query. This concept is very similar to Knowledge Graph, but this term becomes a standard to create this kinds of graph. It is based on HTTP, RDF and URI. RDF is used to describe tuples, URI is used to locate resources, HTTP is used as the transport protocol. Following this publishing standard, there are some datasets existed: DBpedia, FOAF, GeoNames, UMBEL, Wikidata. DBpedia is extracted from Wikipedia, has almost 10 thousand tuples in 11 languages. FOAF mainly describe the relation between people. GeoNames has >= 7,500,000 geometric information.

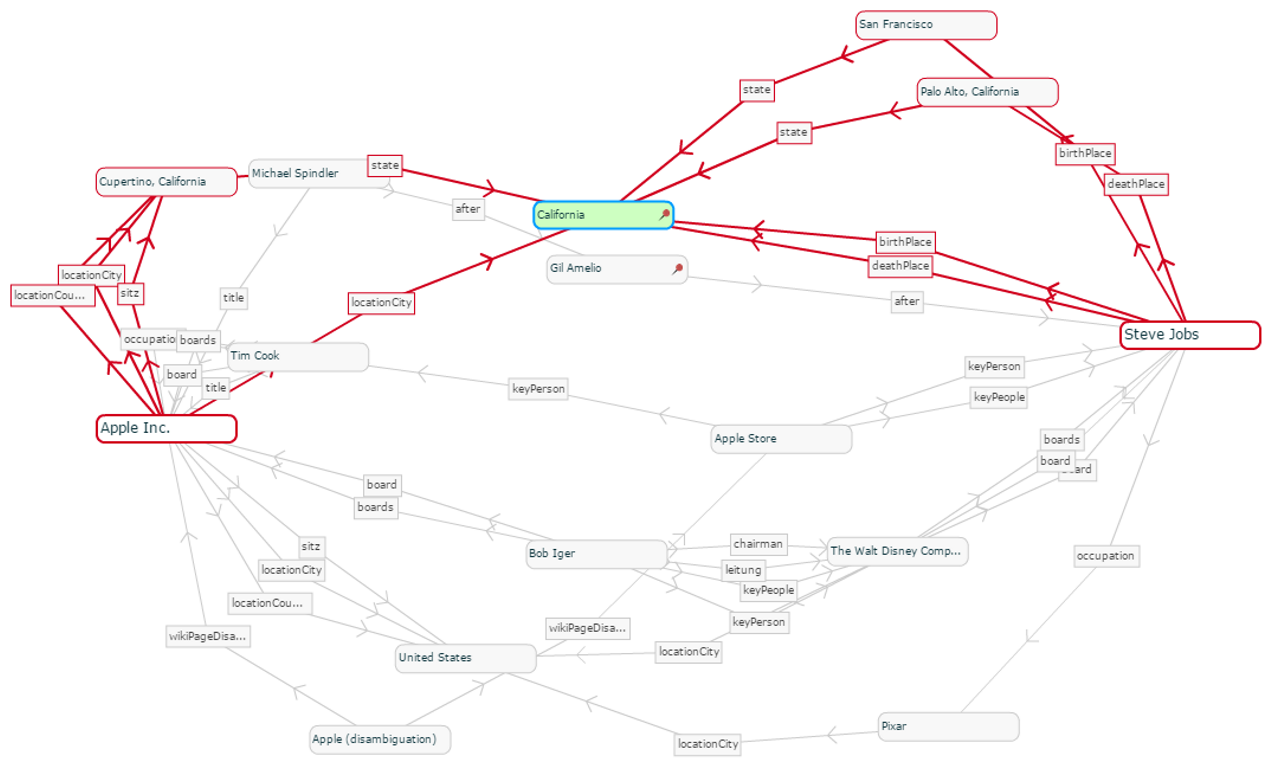
Another good data set is called YAGO. Yet Another Great Ontology. It is extracted from Wikipedia (e.g. categories, redirects, infoboxes), Wordnet (e.g., synsets, hyponymy), and GeoNames. It has need linked to DBpedia ontology and to the SUMO ontology. It provides dumps in Turtle or TSV forms. It also provides query endpoint (Browser or SPARQL).

Ontology is a term which is similar to the namespace in C++. Each field has its own ontology. To separate the knowledge into different fields, we use ontology to describe the aggregation of different knowledge.

Entity Cube is a good relation searching engine build by Microsoft. It is embedded in Bing. When I search ‘薄熙来’ (In Chinese), I can get terms separated into three categories: People, Locations and Organizations. From my perspective, these outs are highly related to ‘Bo Xilai’. But the problem is they are just been put into a big category, which lose the more exact relationship to the target entity. For example, ‘Bo Guagua’ is in the category of People, but it doesn’t show the ‘son’ relation. This is the case when I search in Chinese. The result is better if I search in English. For example, input ‘Bo Xilai’ (In English), ‘Bo Guagua’ is marked the ‘son’ of ‘Bo Xilai’. But the final problem is, it only supports the relations in English corpus. So if I want to leverage this Entity Cube, I have to do a lot of translation work. Which is another hard part of NLP.

* 1. Challenges

Most of the graph reasoning is based on the linear relation chains between two entities, as the Figure Y shows. It illustrates the linear logical path between the entity ‘Steve Jobs’ and the entity ‘Apple Inc.’. They both have one common entity ‘California’, which is the birth/death place of ‘Steve Jobs’ and the location city of ‘Apple Inc.’. People can tell the common part between ‘Steve Jobs’ and ‘Apple Inc.’ easily from this graph. People can say, these two entities are similar due to the common parts they share.



‘Steve Jobs’ has four paths to ‘California’, while ‘Apple Inc.’ has four paths to ‘California’ too. You may wonder, which path pair is the most important one, or, which path can tell the deepest relation between them. From this aspect, we need to compute a score between different path pairs.

One approach is to compute the similarity. This is useful when two entities are similar in prior, like ‘平西王’ and ‘薄熙来’. The sub paths of each relation graph may share some similarity, for example, the relation between ‘平西王’ and ‘重庆’ and the relation between ‘薄熙来’ and ‘重庆’ may be similar. We can even make the constraints looser, such as allowing the end points of two paths are not the same. We can make graph bigger, adding phonetic and spelling relations to the entity.

Besides the similarity between paths, we can also compute the similarity between the sub graph. For graph A = (E1, (R11, E11), (R12, E12)) and graph B = (E2, (R21, E21), (R22, E22)), where E represents the entity node, R represents the relation node. If R11 is similar to R21, R12 is similar to R22, the relation between E11 and E12 is similar to the relation between E21 and E22, then we say these two graphs are similar. This kind of similarity definition can extend the expression ability of the relation graph.

The similarity can help explaining the originating relation between the morph and its target entity, besides the traditional linear logical chains. It can be easily adopted by human beings because we learn new knowledge from comparison. It two items show their similarity, then we can better understand why they are related.

But to define how two relations are similar is very difficult. First, concept has many ways of expression, like the word ‘胜利’, we have a lot of synonyms, like ‘取胜’, ‘打倒’,’摆平’,’获胜’,’成功’,’成就’. They have high similarities, but they are different in characters. Also, they are different in word senses, some are verbs, some are nouns. What’s more, even they are similar, we want to know to which extent are they similar. So we need to score the similarity, not just give a Boolean value to tell whether it is similar or not. We can compute the editing distance, also we can compute the similarity between the context they occur. Second, the noise of the data may interfere the accuracy of the similarity analysis. The data of the relation graph is extracted from web documents like Wikipedia. The content from it are all man-made, so some of them may not follow the format. Also the extraction framework also affects the quality of the information retrieved. Bad format of data may result in the failure of the detection of similarity or the failure of filtering out nonsense similarity. The previous one is called ‘False Negative’, the second one is called ‘False Positive’. For example, instead of ‘胜利’, it is ‘胜利了’, then the additional nonsense word ‘了’ may reduce the similarity between it and ‘打倒’. Another example is, ‘图像大小’. This may refer to the image width of the figure of some entity, but too many of entities have this relation. So at the end this relation has a very high score, but it is very common, whose score should be lowered down.