Discovering Lexical-Semantic Relations from the Person Name Entries of a Machine Readable Dictionary

Anna Leslie Hix  
Department of Computer Science  
University of West Georgia  
Carrollton, GA 30118  
ahix1@my.westga.edu

Muhammad Asadur Rahman  
Department of Computer Science  
University of West Georgia  
Carrollton, GA 30118  
mrahman@westga.edu

Martha W. Evens  
Department of Computer Science  
Illinois Institute of Technology  
Chicago, IL 60616  
evans@iit.edu

Abstract*
In this paper we present our research effort in extracting lexical-semantic relations found in the person name entries of a machine readable dictionary namely the Collins English Dictionary (CED). We used ad-hoc parsing and pattern matching techniques in extracting semantic relations from the definition text. The definition text contains both explicit and implicit relations which we have extracted and stored in a database. We have grouped and manually categorized these relations into various types, such as kinship, identity, part-whole, and so on.

Introduction
Semantic relations between words and concepts are useful in representing knowledge. A human, for example, instinctively understands the concept of a car in terms of it containing parts, such as wheels and seats. In representing this concept in a computer we must explicitly establish these relations such as a car has seats, wheels are parts of a car, etc. Without these relations the knowledge representation in a computer is often incomplete. The way information is organized in the dictionary definitions makes machine readable dictionaries rich sources of semantic information. Each dictionary entry contains various types of relations such as the hypernymy (is-a) or the meronymy (part-whole) relations. The attempt here is to identify and extract lexical-semantic relations beyond the common types automatically from the dictionary definitions of the Collins English Dictionary. The advent of question-answering systems has markedly increased the interest in this kind of information. A large percentage of these questions involve person names or other proper names and many of those are phrased in terms of relationships.

Related Research
Hoang, Strutz, and Evens (1993) carried out a preliminary investigation into the defining strategies and the lexical-semantic relations that can be found in the Collins English Dictionary. They focused on a small subset of proper name entries where they manually identified the explicit semantic relations between the head noun in the defining formula and the proper name entity. Although the number of proper names used in their study was small, nevertheless it was pioneering work in this area of research. A detailed investigation in identifying and extracting lexical-semantic relation automatically from the proper name definitions was carried out by Rahman and Evens (2000a, 2000b, 2000c). Paik et al. (1996) implemented proper noun categorization and standardization module for a document detection system called the DR-LINK. They discussed the expansion of group common nouns and group proper nouns to enhance retrieval recall. Morris and Hirst (2004) found that most of lexical relations in text are “non-classical” relations. They defined classical relations to be the relations that are found in WordNet. They analyzed these relations and discussed their consequences for natural language processing. Our research focused on both the classical and non-classical relations that are found in the proper name entries of a machine readable dictionary.

Methodology
In this research we looked beyond the common semantic relations that were extracted in a previous research project involving the head noun and the proper name entity. We parsed the entire definition text and identified the less obvious and implied relations such as kinship, profession, famous_for etc. The first step in a large number of research steps was the isolation of proper name definitions from the dictionary through the use of a custom built pre-processor. In this pre-processor, person entries were further identified through pattern matching via regular expressions, which search for digit patterns based on birth and death dates. In a subsequent step, we identified the distinct phrasal pattern that defined each type of proper name entry. These distinct phrasal patterns, which are known as the defining formulae, reflect the hypernymy relation in most cases. This step also relied heavily on regular expression pattern matching. Grammatical punctuation and keywords, dependent on the relation being searched for, were employed in recognizing changes in topic within the entries. Most often, topic changes within the entries signified a change in relation, though punctuation occasionally signified only the introduction of a new relation, not a complete relation-type change. During further analysis of the definition text through

* This research was partly funded by an award from the National Science Foundation STEP grant (Generating Enthusiasm in Math and Science) No. 0336571.
parsing we looked for other explicit and implicit relations. The words and concepts along with their relations were stored in a database.

Ad-hoc parsing and pattern matching
Throughout the parsing and pattern matching phase, lexical patterns that we call "stop points" were used to help identify information that needed to be pulled from the entry. Recognition of the vast majority of stop points centered on the location of grammatical punctuation, most importantly commas, semi-colons, and periods. If the input program encountered a specific letter or punctuation sequence, it would stop reading. Some of the exceptions to the standard "stop when you encounter a period" code included esp., B.C., A.D., Dr., Mr., and so forth.

Because of the inconsistencies in formatting, we took an inclusive approach with information, sometimes including more information than was needed instead of possibly leaving out information. In order to accommodate this approach, further splitting and editing tools were written to remove unnecessary information from what we have termed the "rough parse" files. Notably, the relations that were formatted in the pre-parsing process to fit a specific lexical pattern (e.g., FATHER_OF, MOTHER_OF, BROTHER_OF, etc.) rarely required any form of post-rough-parse editing. This realization has led to possible ideas about how to improve the pre-parsing programs. However, much of the superfluous information included in the relations requiring editing and splitting involves implied knowledge.

One relation that simply could not be programmed efficiently was the implied NOTED_FOR, especially when the accomplishments were not preceded by any phrase or word that occurred more than occasionally. These relations were parsed manually, as it was determined that in order to write a program to perform such a task, there would be numerous exceptions that the program would have to check for when parsing. Instead of attempting this time-consuming and inefficient task, it was decided to simply locate each occurrence of this relation and create a file by hand.

While punctuation was most often used to tell when the program should stop reading the entry, key phrases were used to determine when the program should start reading. This approach most often used verbatim phrases; if the relation was looking for authors, each entry would be searched for the phrase "author of." Notable exceptions to this approach included LIVED_DURING, IS_A, and RECEIVED_A. Additional approaches were taken in an attempt to gather as much information from the entries as possible. For example, at least two separate COMPOSER_OF-related files were created using all the methods--one which looks for the phrase "composer of" or "composed" and the other which looks for the phrase "wrote" in conjunction with the occupation "composer." A method written later combined these two files into one master list for the COMPOSER_OF relation.

One of the more surprisingly difficult relations to parse was LIVED DURING. Whereas the other two birth-death relations (BORN_IN and DIED_IN) could be parsed literally from the entries, LIVED_DURING had to locate a specific pattern of digits. Often, entries also included dates in which someone held a certain career or position, and this added problems when trying to locate the correct year span. Below is an example:

Albert I, n c. 1255-1308, king of Germany (1298-1308).

LIVED_DURING had to be able to recognize the 1255-1308 time-span whereas, the 1298-1308 time-span of his reign had to be recognized as RULED_DURING relation. Additionally, the dates often included years expressed in less than four digits. The following examples illustrate the variety in which life spans were recorded:

- 1901-57
- 651-621 B.C.
- 27 B.C. - 2 A.D.
- 954-1002 A.D.
- 1776-1852

Regular expressions were used most often in this process. By creating expressions that would check for a minimum number of digits and a set maximum number of digits in a sequence, along with looking for specific characters following the last digit (i.e., either a comma or a space followed by a capital A or capital B), we were able to determine the lifespan years. Additionally, most other dates (such as the years the person in an entry held a position or career) that did not need to be included were surrounded by parentheses. By instructing the program not to include dates surrounded by parentheses, we were able to include only the information we wanted.

The amount of knowledge that can be pulled from an entry varies with the entry size and depth of information. A smaller entry does not provide very much information, as seen in the following:

Shankar n, Ravi. born 1920, Indian sitarist.

The three relations that can be gathered from this entry are:

- Ravi Shankar BORN_IN 1920
- Ravi Shankar IS_A Indian
- Ravi Shankar IS_A sitarist

In comparison, larger entries provide an enormous amount of information.

Plato n ?427-?347 B.C., Greek philosopher: with his teacher Socrates and his pupil Aristotle, he
is regarded as the initiator of western philosophy. His influential theory of ideas, which makes a distinction between objects of sense perception and the universal ideas or forms of which they are an expression, is formulated in such dialogues as Phaedo, Symposium, and The Republic. Other works include The Apology and Laws.

We extracted the following semantic relations from the above definition:

Plato LIVED_DURING ?427-?347 B.C.
Plato IS_A Greek
Plato IS_A Philosopher
Plato STUDENT_OF Socrates
Plato TEACHER_OF Aristotle
Plato FOUNDER_OF western philosophy
Plato NOTED_FOR influential theory of ideas
Plato WROTE Phaedo
Plato WROTE Symposium
Plato WROTE The Republic
Plato WROTE The Apology
Plato WROTE Laws

Additional information can also be garnered through implied knowledge. Some examples include:

Socrates TEACHER_OF Plato
Aristotle PUPIL_OF Plato
Phaedo WRITTEN_BY Plato
Symposium WRITTEN_BY Plato

This means that a question answering system using our database could answer questions like "Who was Plato's teacher?" or "Name the pupils of Plato."

Recognizing Person Entries
Given the complexity of many definitions within the Collins English Dictionary, the simplest pattern to use when identifying a person entry is a birth-death relation. There are two verbatim birth-death relations, BORN_IN and DIED_IN, and one implied birth-death relation, LIVED_DURING. BORN_IN and DIED_IN can be found literally using the lexical pattern "born" - DP and "died" - DP, where the DP is a digit phrase. LIVED_DURING, however, requires recognition of a specific digit and punctuation phrase, most often resembling "####-####," where each # represents a single digit.

Aga Khan BORN_IN 1933
Hafezal Assad BORN_IN 1928
Nathan Bailey DIED_IN 1742
Cimon DIED_IN 449 B.C.
Clement VII LIVED_DURING 1478-1534
William Cobbett LIVED_DURING 1763-1835

There were 105 entries found using the died_in relation, 2684 entries using the lived_during relation, and 682 using the born_in relation.

People to People Relations
One of the foremost type of relations found were people-to-people relations that describe the relationship between two people. Most often, these were cases of kinship relations (family members), though teacher-student relationships were also found. Interestingly, these relations contained a wealth of implied knowledge. Where an entry states that Person A was the father of Person B, one can quickly infer that Person B is the son of Person A.

John Quincy Adams SON_OF John Adams

There was a plethora of specific relations found, including (along with their specific count of entries):

BROTHER_OF 24
COUSIN_OF 1
DAUGHTER_OF 19
FATHER_IN_LAW_OF 2
FATHER_OF 24
GRANDDAUGHTER_OF 2
GRANDFATHER_OF 4
GRANDSON_OF 10
HUSBAND_OF 10
MOTHER_OF 10
SISTER_OF 3
SON_IN_LAW_OF 3
SON_OF 97
UNCLE_OF 1
WIFE_OF 35
An interesting relation that includes both People to People relations and People to Group relations is DEPOSED_BY. Out of the 9 entries for this relation, 2 refer to groups and 7 refer to people.

Carol II DEPOSED_BY the Iron Guard
Henry VI DEPOSED_BY Edward IV

Part-Whole Relations
Part-Whole relations, for the purpose of this endeavor, describe the relationship between a person and a group of people. This relationship was foremost summed up in the MEMBER_OF relation.

Marian Anderson MEMBER_OF Metropolitan Opera Company
Pete Townshend MEMBER_OF The Who
Francis Poulenc MEMBER_OF Les Six

There were 47 part-whole relations found, with 16 specifically referring to people who were members of a certain family.

John Barrymore MEMBER_OF a U.S. family of actors
Arthur Marx MEMBER_OF a U.S. family of film comedians
Soong Mei-ling Soong MEMBER_OF an influential Chinese family

People to Object Relations
Another type of relation located within the machine-readable dictionary entries is people-to-object. These take on the form of <person> <relation> <an object or objects>. Most often, the object is tangible, such as a book or magazine, though it can also include abstract objects, such as a date.

Rosa Bonheur PAINTER_OF Animals
Felix Bloch RECEIVED_A Nobel prize for physics

Joseph Wright EDITOR_OF The English Dialect Dictionary
Georges Auric COMPOSER_OF ballet and film music

The specific relations found, along with their respective occurrence counts, are:

AUTHOR_OF 311
COMPILER_OF 3
COMPOSER_OF 92
FEAST_DAY_ON 78
INVENTOR_OF 59
PAINTER_OF 24
RECEIVED_A 202
EDITOR_OF 8

To note, the FEAST_DAY_ON relation refers specifically to Catholic saints and their feast days.

Saint Leo I FEAST_DAY_ON April 11
Saint Francis Xavier

The RECEIVED_A relation describes awards and prizes (such as the Nobel Prize, or the Pulitzer Prize) received by the person entities.

Identity Relations
An identity relation identifies a person named entity or shows the identity. The main identity relation is the IS-A. In the Collins English Dictionary it refers to something inherent about a person, such as a characteristic, ethnicity, or even an occupation. IS-A relations are explicit in the dictionary definition and they occur when the definition explicitly defines a named entity to be something, such as in the example below:

Joseph Addison – English essayist and poet

Here we extract the following IS-A relations and the corresponding person-object relationship:

Joseph Addison IS-A English
Joseph Addison IS-A essayist
Joseph Addison IS-A poet

Some other examples are:
Asoka IS_A Indian emperor
Souphanouvong IS_A Laotian statesman
Leopold Stokowski IS_A U.S. conductor
These relations occur in every entry, located in the defining formula segment of the definition text, and they are also easy to extract. Throughout the dictionary, the IS-A relation follows a specific lexical pattern - the ethnicity and occupation of the person always follows the birth-death relation.

Abdul-Hamid II 1842-1918, sultan of Turkey
Johann Pestalozzi 1746-1827, Swiss educational reformer
Saint Valentine 3rd-century A.D. Christian martyr

In each case, by immediately looking past the birth-death relation, one would find the ethnicity and occupation. Interestingly, with occupations and ethnicities, one can find implied knowledge in regards to a PART-WHOLE relation. Consider the definition text for Johann Heinrich Pestalozzi, among the examples above. If Swiss educational reformers is considered the whole, then Pestalozzi can be considered a part, as he held this occupation and ethnicity.

Another interesting identity relation is ALSO-CALLED. This relation can be viewed as a kind of synonomy, since person names designate the same people as do pen names, nicknames, and royal titles, etc. Thus we can categorize these relations as SYNONYMY relations as well. These follow the pattern of relationship followed by second name.

Daniel Louis Armstrong nickname Satchmo
Azor’in pen name of Jos’e Mart’in ex Ruiz
Galen Latin name Claudius Galenus

This relation is quite interesting in that, like the people-to-people relations, ALSO-CALLED provides a wealth of implied information, too. Literally, the relation goes in both directions. For example, where Daniel Louis Armstrong is ALSO-CALLED Satchmo, Satchmo is ALSO-CALLED Daniel Louis Armstrong. With this information a question-answering system could answer the question "What is Satchmo's real name?"

Conclusions

We have shown here that many different types of relations exist in the definition text of proper name entries of the Collins English Dictionary. These different types of lexical semantic relations can be useful in knowledge representation and in inferencing. The relations that we extracted have been categorized manually and we have stored them in a database. We plan to put similar effort into extracting more different types of lexical-semantic relations in other types of entries in the Collins English Dictionary.

Acknowledgments

The authors would like to thank Collins Publishers and Patrick Hanks for their permission to use the Collins English Dictionary for his research. Thanks also to the Data Collection Initiatives (DCI) for providing us with the machine readable version of the dictionary. The initial parse of the Collins English Dictionary was produced by the IITLex group at the Illinois Institute of Technology. Partial funding for this research was provided by the National Science Foundation STEP grant No. 0336571.

References


