Wikipedia as a Language Resource

3.1 Introduction

I imagine that many people make use of Wikipedia, the world's largest online encyclopedia. Although issues with its reliability have been pointed out, I use it to look things up on a daily basis due to the vast amount of information that can be obtained from the greatest encyclopedia of all time in terms of both in quality and quantity. Also, as I am involved in researching Wikipedia as social data, I am extremely interested in its pageview count statistics.

There seem to be many researchers who treat Wikipedia as a subject of their research. For example, DBpedia*[1] and its Japanese version*[2] are endeavors that release data they have extracted from Wikipedia and converted to LOD (Linked Open Data) format files, which are apparently used for research into the Semantic Web among other things. Wikipedia is also recognized in the natural language processing research field as a language resource storing vast quantities of sample sentences, and a variety of methods for utilizing it have been proposed. Consequently, in this report I would like to step away from my own research and discuss the theme of Wikipedia as a language resource.

3.2 Writing "Unix Archaeology"

One of the reasons I became interested in Wikipedia as a language resource goes back to when I published a book called "Unix Archaeology*[3]" in April of this year. This book introduces historical facts related to UNIX based on various documents, and if pressed I'd have to categorize it as a history book.

This book is based on a collection of 26 articles that were published in the monthly magazine "UNIX USER" between 2003 and 2005, now organized and re-edited in book form. One of its characteristics is that it was written purely using reference material collected through Google Search, without any interviews or trips to the library. I don't know about today, but back then only a few years had passed since Google's search engine has been released in 1998, so this was a rather reckless writing style.

That said, at the time I had faith in this approach (which thinking back now was based on extremely flimsy grounds). This relates to an occupation that appeared in the 1980s called a "database search engineer," also commonly called a "searcher." A TV program covering this job discussed the example of a novelist and searcher teaming up to write a new novel, in which the searcher told the novelist that they didn't need to gather any reference material at all to write the story, further boasting that if the novelist told them the information they needed, it could all be pulled out of a database. As a rookie who had just entered employment, at the time I was very skeptical that such a thing could be possible. When I was asked to write a series of articles more than ten years later, I remembered this episode and thought this approach may be possible for me as well, with Google's search engine now at my disposal.

In reality, there was a delay of about six months between the time I accepted the writing request and when the first article was published. The editorial department expected me to use this time to accumulate several articles worth of material, but I spent most of the time looking into keyword selection and sorting order, or in other words search patterns, to locate the material I wanted using Google. As a result, six months later I had only completed the first and second articles, and afterwards I had to pay fairly stiff reparations. This is how I ended up settling on a slightly eccentric writing style, adding to my manuscript under the topic of "delving as deep as possible" with search engine in hand.

3.3 Drilldown Searches

Now there is a term called "drilldown" that perfectly expresses "delving as deep as possible." According to Wikipedia's "Drill down"*[4] entry, this is defined as to move from one place to another, from information to detailed data, by focusing in on something."

*[1] DBpedia (http://wiki.dbpedia.org/).

*[2] DBpedia in Japanese (http://ja.dbpedia.org/).

*[3] Unix Archaeology (http://asciidwango.jp/post/142281038535/unix%E8%80%83%E5%8F%A4%E5%AD%A6-truth-of-the-legend) (in Japanese).

*[4] Drill down (https://en.wikipedia.org/wiki/Drill_down).

I imagine for me the illustrations given on this page regarding online users and web-surfers are the closest match, but my focus was on digging up documents that weren't well known regarding the history of computers, which I'd say was a very special case.

For example:

> In 7th Edition UNIX, the kernel was also completely rewritten in C. This is a relatively well-known fact. From a number of documents left by Dennis Ritchie (papers, lecture materials, and interviews), we can confirm that this task was undertaken by Ken Thompson, Dennis Ritchie, and Steve Johnson, and that Steve Johnson was involved because he was the developer of the Portable C Compiler (PCC). So, are there any documents in which Steve Johnson discusses the development task himself?

Alternatively:

> Regarding the development of BSD UNIX, when 4BSD was released there was criticism mainly centered around the fact that it performed worse than 3BSD, and to deal with this the UCB's Computer Systems Research Group released 4.1BSD with performance tuning. This is brought up in many documents discussing the history of UNIX. However, a document authored by Kirk McKusick revealed the fact that this tuning was done by Bill Joy, who was infuriated by an extremely combative critical article by a person called David Kashtan. So, are there any documents that discuss the content of this critical article by Kashtan, as well as the counterargument by Bill Joy?

When attempting drilldowns related to historical facts like these, searches are extremely difficult to carry out. Repeating the cycle of reading the relevant parts of documents you've found and selecting keywords from within these to search for new documents, and then writing a six to ten page article every month, is a task that requires a lot of energy and concentration. Rather than the literary work of a novelist, I'd liken it more to a journalist writing an article.

In fact, when it was decided the articles would be published in book form 12 years after serialization ended, it was necessary to do a considerable amount of rewrites and compose new text, so I tried to think back to that time and work in the same way again. However, drilling down to historical facts had become an unbearably hard task for someone of my age. I began to wonder whether at least part of this work could be handled using computers.

3.4 Named Entity Recognition (NER)

Since then, I have continued to ponder the computerization of historical fact drilldowns. This was initially more about making preliminary arrangements for my writing activity, rather than for research. Of course, there was also the ulterior motive of it lowering the hurdle for receiving writing requests (laughs).

To build software that computerizes this process, or in other words mimics the way I search for documents, it is first necessary to clarify how I had been locating them. The most important knowledge regarding bibliographic searches and the extraction of factual information using a search engine that I picked up through writing the aforementioned articles was to focus on proper nouns. It may seem conventional, but this corresponds to the names of people and organizations, as well as computer model numbers and nicknames in the case of historical facts regarding computers. By collecting as many of these proper nouns as possible and specifying combinations of them as search keywords, it was often possible to greatly narrow down search results in the way that I desired. In light of this, I set to work on finding a technique for extracting proper nouns from English text.

Not surprisingly, this involves the field of research into natural language processing, and straight after I began looking I realized two things.

(1) Statistical natural language processing is mainly used now
(2) The aspects of natural language processing being researched are considerably different between Japan and English-speaking countries.

Regarding the first point, this has completely different objectives and goals to my current research, but in both cases research is carried out using statistics-based techniques. In particular, the basic technology of data analysis is also a subject I research, and looking just at this basic technology we can see there are many common points.

Regarding the second point, until then I had in some respects assumed that when the subject of natural language processing came up, it implicitly referred to Japanese. I thought the main theme was areas of research closely associated with Japanese, such as morphological analysis and machine translation. But I now get the impression that English natural language processing (understandably) incorporates very different themes.

The technique of extracting proper nouns from English text I sought turned out to be one of the main themes of natural language processing in English--a process called Named Entity Recognition (NER).

3.5 NER in the Natural Language Toolkit (NLTK)

Today, NER is incorporated into the Natural Language Toolkit (NLTK), so as shown in Figure 1 you can attempt to extract named entities comparatively easily using Python script.

When given an English text file name, this script executes NER and produces the kind of output shown in Figure 2.

It displays a word, part of speech, and NE tag on each line. When the first letter of the NE tag is “B-” it means the word is the first part of a named entity, whereas “I-” is a continuation of one. PERSON and ORGANIZATION are self-explanatory, but GSP stands for “Geo-Socio-Political group.” We can see that “Hillary Clinton” and “Donald Trump” are recognized as people’s names.

Next, as an example of a slightly larger body of text, I tried extracting people’s names from the “Twenty Years of Berkeley Unix” book by Marshall Kirk McKusick, who also aided me in writing Unix Archaeology (Figure 3).

There is some misrecognition, but in addition to well-known names such as Ken Thompson, Dennis Ritchie, and Bill Joy, BSD UNIX development staff members other than Kirk McKusick, such as Ozalp Babaoglu, Sam Leffler, Mike Karels, and Keith Bostic, have also been correctly recognized. This demonstrates that recent research into statistical natural language processing has made it possible to extract named entities with a certain degree of accuracy, without any special tuning for a particular purpose (such as topics related to the history of UNIX).

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import nltk
import sys

if __name__ == "__main__":

    param = sys.argv

    args = param[1]

    with open(args, 'r') as f:
        sample = f.read()

    sentences = nltk.sent_tokenize(sample)
    tokenized_sentences = [nltk.word_tokenize(sentence) for sentence in sentenc\es]
    tagged_sentences = [nltk.pos_tag(sentence) for sentence in tokenized_senten\ces]

# for NLTK 2.7
#   chunked_sentences = nltk.batch_ne_chunk(tagged_sentences)

# for NLTK 3.0
    chunked_sentences = nltk.ne_chunk_sents(tagged_sentences)

    entity_names = []
    for tree in chunked_sentences:
        print nltk.chunk.tree2conllstr(tree)
```

Figure 1: named_entity_recognition.py

3.6 Statistical Natural Language Processing and Corpora

It would probably be necessary to study research into statistical natural language processing to learn the inner workings of how it is possible to achieve decent extraction accuracy without any special tuning. However, "corpus" is a term that crops up frequently when reading reference material.

According to the English Wikipedia, in linguistics a corpus is "a large and structured set of texts," which is "used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory." To an outsider like me this explanation is hard to comprehend by itself, but if you remember Zipf's law many people are familiar with that states the percentage of the total accounted for by a word with an appearance frequency of rank k is proportional to 1/k, you may find it a bit easier to visualize the mysterious relationship between our writings and statistics. In other words, when people write compositions they tend to do it with an unconscious statistical bias. I imagine this is why statistical natural language processing works so well.

I actually experienced this behavior with a statistical bias viscerally through writing "Unix Archaeology." For example, Dennis Ritchie published a large number of documents that discuss the development background of UNIX on his home

```
$ cat NHK-short.txt
A US poll shows that Democratic presidential nominee Hillary Clinton's lead
over her Republican rival, Donald Trump, has shrunk to 3 percentage points.
$ ./named_entity_recognition.py NHK-short.txt
A DT O
US NNP B-GSP
poll NN O
shows VBZ O
that IN O
Democratic JJ B-ORGANIZATION
presidential JJ O
nominee NN O
Hillary NNP B-PERSON
Clinton NNP I-PERSON
's POS O
lead NN O
over IN O
her PRP$ O
Republican JJ B-ORGANIZATION
rival NN O
, , O
Donald NNP B-PERSON
Trump NNP I-PERSON
, , O
has VBZ O
shrunk NN O
to TO O
3 CD O
percentage NN O
points NNS O
. . O
$
```

Figure 2: Results of Executing NLTK's Named Entity Recognition on a Sample English Passage

```
Alan Nemeth
Babaoglu
Beranek
Berkeley
Berkeley Software Design
Berkeley Software Distribution
Berkeley Unix
Bert Halstead
Bill Jolitz
Bill Joy
Bob Baker
Bob Fabry
Bob Guffy
Bob Kridle
Bostic
Casey Leedom
Chuck Haley
DARPA
Dan Lynch
David
Dennis Ritchie
Dickinson R. Debevoise
District Judge
Domenico Ferrari
Duane Adams
Eugene Wong
Fabry
Fateman
Ferrari
Freely Redistributable Marshall Kirk McKusick Early History Ken Thompson
Haley
Hibler
Jeff Schriebman
Jerry Popek
Jim Kulp
John Reiser
Jolitz
Joy
Karels
Keith
Keith Bostic
Keith Lantz
Keith Standiford
Ken Thompson
Laura Tong
Leffler
Linux
Lucasfilm
Math
Michael Stonebraker
Mike
Mike Karels
Mike Muuse
Networking Release
Ozalp Babaoglu
Pascal
Pauline
Peter Kessler
Professor Domenico Ferrari
Professor Richard Fateman
Professors Michael Stonebraker
Ray Noorda
Rick Macklem
Rick Rashid
Rob Gurwitz
Robert Elz
Sam Leffler
Schriebman
Schwartz
Statistics
Support Meanwhile
Susan Graham
System
System III
System Manual
System V
Tahoe
Thompson
Tom London
Unix
Unix Early
Utah
```

Figure 3: Persons Mentioned in "Twenty Years of Berkeley Unix"

page*5 This came in handy while I was writing my book, but upon reading a number of sentences over and over I often noticed identical turns of phrase used in multiple different places. In short, it seems wording that readers may interpret as a habit of the writer can be recognized as a statistical bias through statistical text analysis.

Incidentally, Dennis Ritchie's most noticeable habit was to write the name of the developer of PCC as "Steve Johnson." Ritchie's writings all use this name, but the developer's real name is Stephen Johnson, and in his own papers and Wikipedia entry the spelling Stephen is used. I'm not sure whether this is a misunderstanding by Ritchie, or if the developer was known by the nickname Steve at Bell Labs, but either way it baffled me a lot when I was trying to follow up facts and found no documents at all no matter what criteria I searched for.

It seems that statistical natural language processing is a form of research that obtains a variety of new knowledge from the statistical bias in text written by people.

3.7 Creating a Named Entity Corpus from Wikipedia

Let us go back to discussing NLTK. NER in NLTK is outlined in "Extracting Information from Text" in the seventh chapter of "Natural Language Processing with Python," and its implementation is the "ACE Named Entity Chunker (Maximum entropy)" listed first in the NLTK Corpora*6.

This machine learning model called "Maximum entropy" (there is an explanation under "6.6 Maximum Entropy Classifiers" in the same book) uses an NE corpus with named entity tags. The creation of an NE corpus with named entity tags such as "PERSON" or "ORGANIZATION" is an extremely time consuming task that must be done manually, and for this reason almost no NE corpora are distributed for free. In fact, for NLTK as well only the chunker trained on the Automatic Content Extraction (ACE) corpus is distributed as a pickle file. The corpus itself is not included.

There have been efforts to automatically generate this NE corpus using Wikipedia page data. The paper entitled "Transforming Wikipedia into Named Entity Training Data*7" proposes the use of Wikipedia to create named entity tagged corpora.

The terminology and names found in Wikipedia articles are often linked to other related articles. The basic idea of this paper is to convert these links between articles into named entity tags. For example, in an article about the character "James Bond" in the novel "Thunderball" by "Ian Fleming," we can expect each named entity to have mutual links set up between them. The Ian Fleming article would no doubt indicate that this named entity is a person, and the Thunderball article would show that it is a novel. In other words, it is possible to tag the original article automatically by tracing the links (Figure 4).

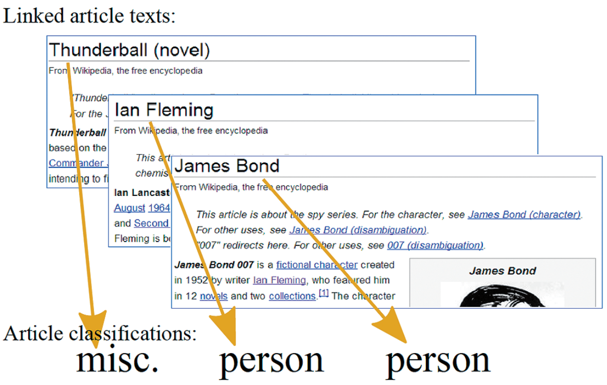
Linked article texts:
Thunderball (novel)
From Wikipedia, the free encyclopedia
"Thunde
For the
Thunderball
based on the
Commander
intending to fi
Ian Fleming
From Wikipedia, the free encyclopedia
This art
chemis
Ian Lancast
August 1964
and Second
Fleming is be
James Bond
From Wikipedia, the free encyclopedia
This article is about the spy series. For the character, see James Bond (character).
For other uses, see James Bond (disambiguation).
"007" redirects here. For other uses, see 007 (disambiguation).
James Bond 007 is a fictional character created in 1952 by writer Ian Fleming, who featured him in 12 novels and two collections.[1] The character
James Bond
Article classifications:
misc.
person
person

Figure 4: Deriving Training Sentences from Wikipedia Text

The paper states it is possible to create a massive corpus by extracting millions of articles from Wikipedia for NER training. The following four steps are given for this process.

1. Classify all articles into entity classes
2. Split Wikipedia articles into sentences
3. Label named entities according to link targets
4. Select sentences for inclusion in a corpus

*5 Dennis Ritchie (https://www.bell-labs.com/usr/dmr/www/).

*6 NLTK Corpora (http://www.nltk.org/nltk_data/).

*7 Transforming Wikipedia into Named Entity Training Data (https://www.aclweb.org/anthology/U/U08/U08-1016.pdf).

Using this procedure, in the paper an attempt is made to create an NE corpus based on the standard CoNLL categories of entity class (LOC, ORG, PER, and MISC).

Applying this procedure would clearly be a big data process in the case of Wikipedia, which currently has over five million articles in English and over a million articles in Japanese. In particular, the bootstrapping approach to classification indicated in the paper, which involves some manual work to identify classes, is a tricky issue to deal with when you want to create a corpus by machine alone.

3.8 Conclusion

Unexpectedly, I get the sense that the drilldown searches for historical documentation that I imagined would progress quite far if I applied the findings of the latest natural language processing research. I am also a little surprised that the basic technology developed for the analysis of Wikipedia that I am involved with in my research could also be used for this initiative.

The task of extracting the names of people and systems from documents could be achieved using NER, but to improve the accuracy of this it will be essential to obtain NE corpora and use these for training. It was good to learn that the Wikipedia we utilize day-to-day can be used as a resource for creating our own NE corpora. The remaining issue is figuring out how to pick up articles on people from the English version of Wikipedia, which has over five million entries.

I don't see this issue as a big problem for the history book writing I am engaged in right now. This is because the writing of a history book is nothing but digging up information on eras, people, and events based on a theme the writer determines. The task of collecting articles from the English Wikipedia on people associated with the theme I am writing about is a common daily practice for me. Reporters that write non-fiction or news stories and social science researchers have thorough knowledge of named entities other than people, including their classification. I believe considering ways to gather and share their collective wisdom would be the quickest way to resolve problems.

That sums up how an unforeseen situation led to me learning about the current state of natural language processing research. I think it will be possible to apply the results of this to the analysis of social data that I am handling in my research right now as well. One example would be "factorial analysis." Performing time series analysis of social data enables you to detect sudden fluctuations or bursts, but to investigate the root cause of these you need to trace the social trends that were in effect at the time. I had considered collecting new stories and other data using a search engine, and I think the methods I have introduced in this report could be used to determine the search criteria for achieving this.

Author:
Akito Fujita
Chief Architect, Business Strategy and Development Center, IIJ Innovation Institute Inc. (IIJ-II). Mr. Fujita joined IIJ in 2008.
He is engaged in the research and development of cloud computing technology utilizing knowledge gained through structured overlay research.