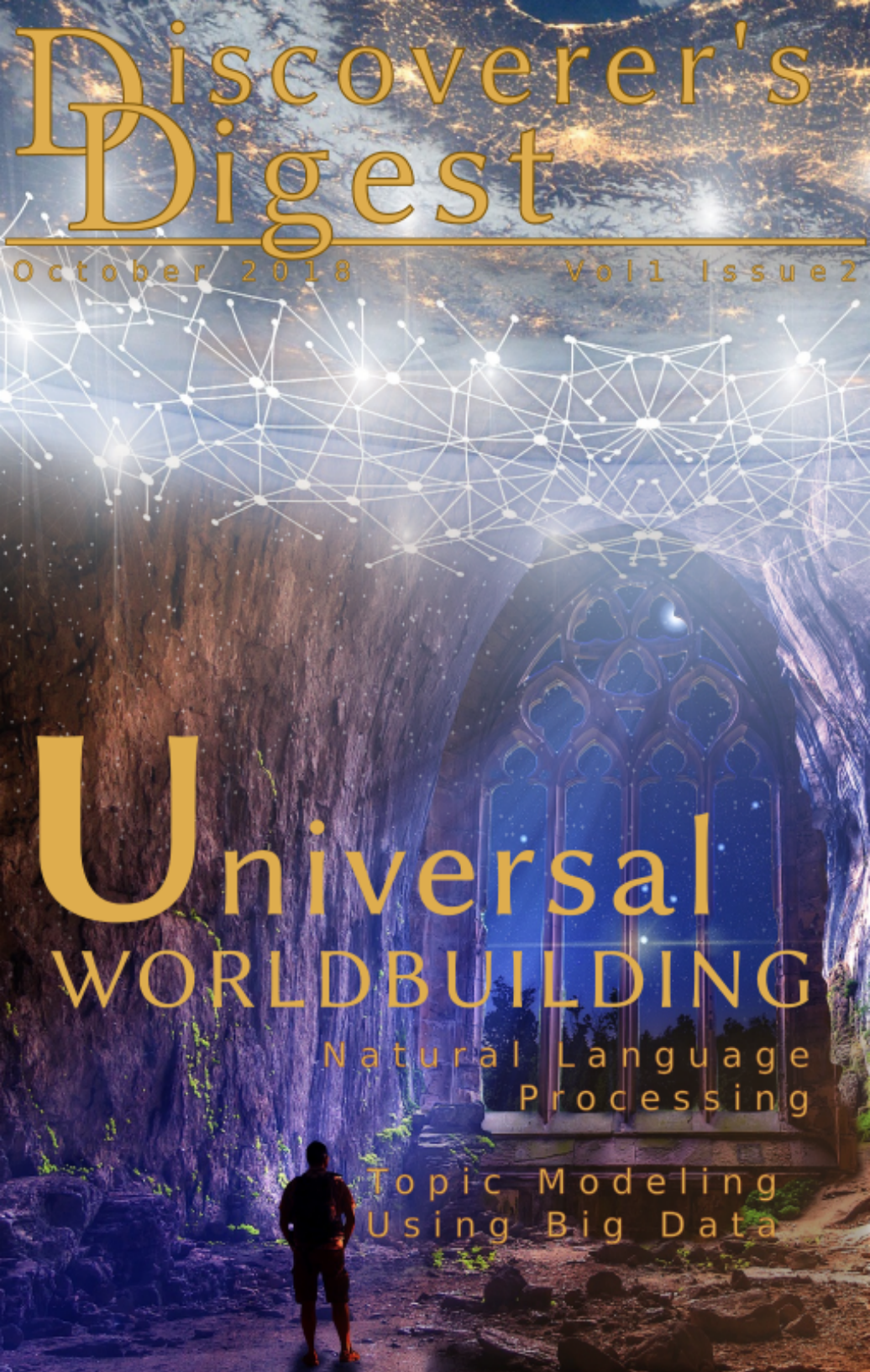
Discoverer's Digest

October 2018 Vol1 Issue2



Universal WORLDBUILDING

Natural Language Processing

Topic Modeling Using Big Data

Contents

Credits

Creator / Graphics / Author: D. Cooper Stevenson

Cooper Stevenson dedicates himself to enabling fulfilling human experiences through the aesthetic use of technology. He lives with his wife, *Deanna* and his two cats, *Tiberius* and *Chester*.

Editor: Peter M.J. Grose

Peter is an editor and online content specialist who has been entertained by interactive fiction since *"Nord and Bert Couldn't Make Head or Tail of It."* He had to resort to a walkthrough in order to complete *Trinity*, and he could never figure out *A Mind Forever Voyaging*.


New
HORIZONS
Aritificial Intelligence
Engaging
World
Models

I

New Horizons

Artificial Intelligence For Engaging World Models

WE NOW HAVE effectively limitless capability to build dynamic, immersive world models.

"Any sufficiently advanced technology is indistinguishable from magic."

–Clarke's Law

Gary Kasparov's match with "Deep Blue" was highly publicized and carried philisophical implications

In 1997 champion Gary Kasparov lost to IBM's "Deep Blue" supercomputer in a highly publicized match. In 2011, IBM's purpose–built computer, named Watson and developed to compete on the quiz show *Jeopardy!*, prevailed over legendary champions Brad Rutter and Ken Jennings to win first place and take the $1 million prize. In 2016, Google's AlphaGo computer program defeated the Go world champion, Lee Sedol, in a five–game professional match. In each case, the computer scientists is met with skepticism. In each case, the dedicated technologists proved the skeptics wrong. At the crest of each new achievement, philosophers asked the next logical questions thus restarting the cycle of discovery and progress.

Editor's note: margin notes in red signify links pointing to sites of reference

Wikipedia: AlphaGo

Watson prevailed over legendary champions Brad Rutter and Ken Jennings to win first place and the $1 million prize

By leveraging big data, artificial neural networks, and natural language processing we can continue to explore new horizons.

SOCRATIC METHOD OF IMMERSION

What makes games so engaging is their way of pushing our human experience. They patiently and quietly ask us to tackle tough problems on our own. They make us search our knowledge and intuition. The quiet rewards for our struggles are those moments in our lives when we feel like we accomplished the before unattainable. Challenging, positive interactivity engages our souls. It makes us, "better than yesterday."

Spock's formative schooling in "Star Trek" depicts hyper–interactive education

If engagement is a significant factor in quantifiably productive experiences then it stands to reason that a kind of Socratic "give and take" yields higher satisfaction and knowledge retention. Interlocutors, like Kasparov or Sedol, query their environment. It replies with meaningful responses. Similarly, the interactive question–and–answer format of *Jeopardy!* offers a distilled form of mutual concession; it's mechanics lay bare the underpinnings of good Interactive Fiction. When we ask intelligent questions of our world we expect fulfilling, logical answers. When we get meaningful answers, we treasure that meaning like a prized possession.

Lectures aren't just boring they're ineffective too, study finds

MAKE IT GOOD

Chris Crawford describes the value and nature of good interactivity:

Chris Crawford expresses the nauture and value of interactivity in an interview for "Get Lamp"

> The value of interactivity arises from the way the human mind responds to it. The human mind is not a passive receptacle—it's an active agent. We don't learn by sitting on our fat butts and listening or watching or hearing or passively receiving information. We learn by reaching out to the world and messing around with it with our fingers.

The Role of Interactivity in Student Satisfaction and Persistence in Online Learning

Experience and research shows that high–quality interactivity enhances the participants' satisfaction and persistence with the content. But the interactive environment must remain focused, substantively complete, and—most of all—engaging. Understanding the power and fragility of interactive models means understanding that the construction of solid frameworks is necessary from the very start of our work.

Complete interactivity *prima facie* implies that we need to build a complete world model. By our "completeness theory," the reader must be able to rob a bank in a work of interactive romance comedy, or else the system cannot provide plausible engagement. Fortunately, experience suggests that total world modeling is not necessary for immersion. We find our saving grace in the capability of our minds to suspend disbelief for the purpose of enjoying the experience.

In an interview for *Get Lamp*, Crawford addresses the level of detail required for our world model to provide credible stories:

Chris Crawford discusses the extent to which a complete world model is necessary in an interview for "Get Lamp"

> A lot of people, I think, mis–understand the nature of interactivity thinking that it requires giving the player absolute freedom. That's not at all what interactivity requires. It merely requires giving the player *all relevant, useful freedom.*

Audiences consume themselves in stories while sitting on stone slabs through suspended disbelief

We've seen audiences enjoy and learn from stories from as far back as the Greek Tragedies, when great warriors and fair maidens entertained crowds that sat in stadium style theaters on stone slabs. A story's efficacy doesn't come directly from the physical setting of the audience. The theater sets the mood, while the careful crafting of dialog, executed with superb timing, delivers the story's impressions, which can be profound. The message speaks to us on primal levels

that can be measured. Beyond the science, writers deliver experiences by relying less on settings and more on *story*.

Paul J. Zak, How Stories Change the Brain

THE STRUGGLE ETERNAL

Writing complete Interactive Fiction worlds by hand is a long, arduous process. There's no way around it. Aaron Reed, in an interview for *Get Lamp* notes,

> As a player of Interactive Fiction, it's very easy to not appreciate the work that went into it. You just come across one error message or one response that the designer didn't think of and you think, "Oh, well, forget this game." After you've actually gone through and had to think of all those things and code them and write them yourself, your perspective really changes.

Aaron Reed notes the complexities of creating a complete world model in IF

Complete world models, on the other hand, where the Interlocutor feels confident that their interactions will be met with meaningful responses is known to be the most important factor in their enjoyment of Interactive Fiction. Emily Short writes in her article, "The Prose medium and IF:"

> By contrast, negative points made about IF in the wider world (and here I draw from threads on SLASHDOT about the yearly comp, and discussions on various gaming boards) often focus on the parser, not the writing, as the critical weakness of IF. It is the refusal to recognize nouns mentioned in the text, the insistence on very particular player behavior, the guess-the-verb nonsense, that turns off the most people, as I observe it.

Emily Short considers the reltionship between prose and parser function in, "The Prose medium and IF*"*

So it is that precisely what the reader finds most important to a good work of IF turns out to be the most burdensome for the implementer to write. The implementer's task isn't limited simply by sheer mass of the world's components; his undertaking is compounded in difficulty by the fact that he must think of everything a reasonable reader will try to do in the world for any given situation. The player's complete agency and the implementer's difficulty furnishing his world model mix like oil and water.

CRITICAL NON–ACCLAIM

Some argue the readers expectation of agency and the implementors burden to achieve the goal will never resolve. They will say that there is no way an implementor will come even close to a satisfactory world model without resorting to trite gimmiks. Proponents of this argument point to efforts in the past, most of which have fallen short. They will pine eloquent about the absolute complexities of the task's implied workload and call upon the implmenter as running a fool's errand.

To the critics I respond as Theodore "Teddy" Roosevelt did,

> It is not the critic who counts; not the man who points out how the strong man stumbles, or where the doer of deeds could have done them better. The credit belongs to the man who is actually in the arena, whose face is marred by dust and sweat and blood; who strives valiantly; who errs, who comes short again and again, because there is no effort without error and shortcoming; but who does actually strive to do the deeds; who knows great enthusiasms, the great devotions; who spends himself in a worthy cause; who at the best knows in the end the triumph of high achievement, and who at the worst, if he fails, at least fails while daring greatly, so that his place shall never be with those cold and timid souls who neither know victory nor defeat.

I argue that, given today's IF authoring technology to works of the past is a little like comparing the Nicéphore Niépce's first fixed image with a camera to today's moving experiences found in modern film.

We have virtually no limit on the size and scope of our work and we can use all available mediums in the world model through scripting hooks available with both TADS3 and INFORM 7. I submit boldly that it ought even be possible using projectors to let players roam around in a physical building or geodesic dome where the rooms, using projectors, change as the story does.


I Enjoy
sunsets and
long crawls down
dank cave passages
Discoverer's Dating Service
(800) 867-5309

CRITICAL CONSTRUCTS

The question of creating rich, virtual experiences quickly escalates into varying narrative techniques and structures, fluid linguistics, and other aspects. I will happily leave these discussions to the scholars as we focus on creating a foundation that they may all build on. Our goal is to build an efficient and flexible framework whose capabilities enable all comers.

The solution, whatever it may be, should answer this core question: "How do we build an engaging world whose dynamics deliver enriching engagement for the audience?" We assume no bounds and take only the requirement for verifiable results to achieve our goal. If our work passes a literary Turing Test, we are successful. Hopefully (and probably), we've created a framework for others to follow and develop further. Our story's positive effect on the audience is our mission.

Wikipedia: Touring Test

To date I know of no substitute for inspired creativity. There exists no system—quantum or otherwise—that serves even as a shallow replacement for inspiration. Our minds possess sophisticated neural methods for spotting contrivance at a glance. We're embroiled in an arms race between lying and honesty detection that continues to this day. And so far the longevity of inspired works shows a direct correlation with honesty; Plato's "The Republic" survives to this day, while last month's "Enquirer" headlines do not.

Rob Brooks, The Evolution of Lying

That is the first plank of our system's foundation: raw inspiration space for the writers expressing their brutal honesty. There's no "man behind the curtain" in our model. Just as Greek theaters made no claim to be a of being Greco-Punic battlefields, we make no claims that our emulated world is a complete manifestation of the real environment. "However," we say, "if you let us, we will take you on a journey of laughter, sadness, or possibly both. Perhaps we will make you think in ways you have never thought before. Above all, we will make you *feel*."

NUTS AND BOLTS

This issue of *Discoverer's Digest* provides step–by–step instructions to build topic modeling and natural language processing systems that

can greatly reduce the effort required to build your world and narrative models. Building an authentic world requires an enormous amount of writing; these techniques can assist greatly by generating ideas and taking care of the mundane. The author may focus on *story* rather than clerical tasks and associations.

Both techniques draw (or can draw) their results from the entire collection of Wikipedia articles (all 5.7 *million* of them, in English alone). It's as if the entire collection human thought was at your disposal to build your work of Interactive Fiction. The result is a systematized method for creating complete, relevant world models.

Topic modeling is a systematic way to flesh out areas that are related to a work's central themes—the broad areas that an Interactive Fiction audience is likely to explore. Topic modeling also helps generate completely new areas of reference we often hadn't thought of.

It provides a ready–made system that suggests the most relevant topics for a query. Those topics can be re–entered into the system to provide greater depth and generate additional detail with more relevant topics, and so on.

The issue also discusses natural language processing, where we select a subject or word phrase to see how it is used in context with surrounding text. It provides brief phrases that span multiple contexts for works of Interactive Fiction.

Both articles are somewhat technical. However, the step–by–step instructions should make it easy to replicate their results; each step is explained along the way. These articles use the LINUX operating system but their techniques may be adapted for other operating systems.

All in all, this issue provides foundational frameworks to build the scope and context of your authorship quickly, assisting with contextual phrasing for each geographic or narrative node in your story. Running your topics and phrases through these systems will provide you with quick "stubs" to support your work. Its suggestions will provide moments where you think, "ah, it's true that x is related to y, I didn't think of that."

I hope you enjoy this magazine. *Discoverer's Digest* seeks to cover all manner of topics related to Interactive Fiction, ranging from parser

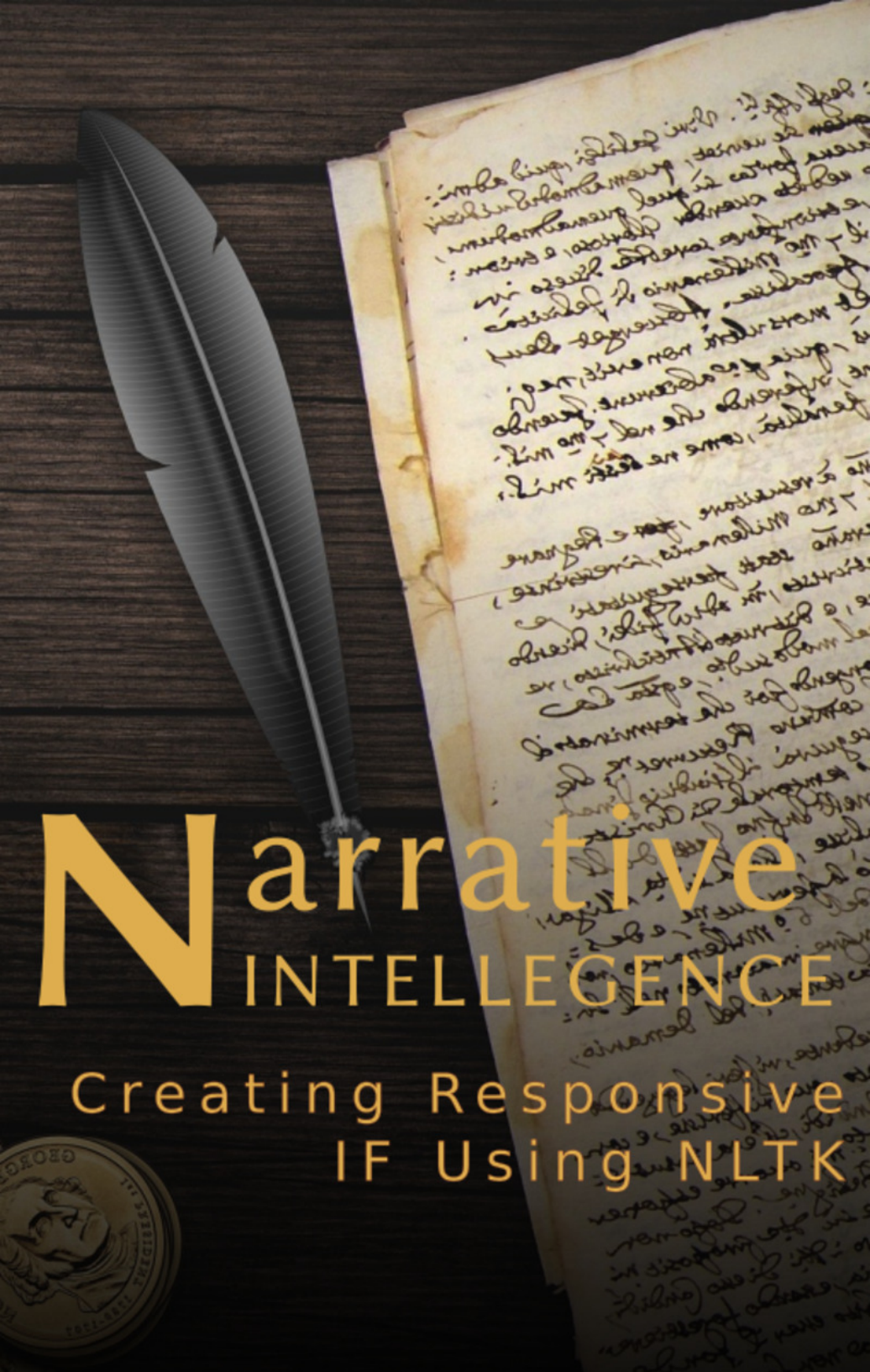
experiences to choice—games, along with experiments in the medium (including General Artificial Intelligence) that we've not yet seen.

And please submit your news, story ideas, and comments by sending a personal email to cooper@cooper.stevenson.name.

Happy Writing!

D. Cooper Stevenson


JOIN
THE
HELLO SAILORS
UNION
LOCAL
220
HELLO SAILORS UNION
"Get Your Own
Damn Boat"
(800) 867-5309


Narrative
INTELLEGENCE
Creating Responsive
IF Using NLTK

II

Narrative Intellegence

Creating Responsive IF Using NLTK

REMOVING THE BURDEN to implement every detail goes a long way toward creating polished IF environments

"If you put yourself in a position where you have to stretch outside your comfort zone, then you are forced to expand your consciousness."

–Les Brown

Natural Language Processing [nltk], may hold the key to breakthrough levels of engagement in if. nltk's power is found through it's ability to query millions of lines of text, retrieve quantified answers found in the source material, and bring to light relationships between entities. The context suggestions, when appropriately applied, translate into actionable intelligence for the author and greater submersion for the reader. Given nltk's ability to retrieve massive numbers of contexts and relationships in seconds make nltk techniques powerful instruments for building immersive world models.

Text analysis with nltk cheatsheet

nltk is not only fast it's architecture lends itself to automation. Once the author lays down the topics to be covered (likely using topic modeling as described beginning on page 25) an article summerizor is then used to pull out each topics' salient features output by a few sentances. These results are then fed into nltk for context, parts–of–speech, and related associations specific to each result. Natural language processing with nltk modular architecture provides volumes of relevant results and may be molded into a workflow that works best for you.

Responses for virtually all kinds of queries to the world model may efficiently built using nltk's ability to bring about the following features:

- Varying, sometimes wide–ranging contexts for a particular topic or object
- Brings forward parser responses you may not have considered
- Tags objects mentioned in the main prose to avoid "I don't see any x here" here messages

A TIME AND A PLACE

Corpos: nothing more than a large body of text, sometimes referred to as, "a big bag of words"

Suppose we're writing a character modeled after Captain Ahab in the seminal classic, "Moby Dick." After loading the Moby Dick corpus, included with nltk, we query the system for instances in context where Captain Ahab appears:

```
 >>> text1.concordance('ahab')

, eh ? Have ye clapped eye on Captain Ahab ?" " Who is
    ↪ Captain Ahab , sir ?" " A
e on Captain Ahab ?" " Who is Captain Ahab , sir ?" "
    ↪ Aye , aye , I thought so .
 " Aye , aye , I thought so . Captain Ahab is the
    ↪ Captain of this ship ." " I am
ast backing out . Clap eye on Captain Ahab , young man ,
    ↪  and thou wilt find that
ptain Peleg , inquiring where Captain Ahab was to be
    ↪ found . " And what dost tho
 " And what dost thou want of Captain Ahab ? It ' s all
    ↪ right enough ; thou art
l thee . He ' s a queer man , Captain Ahab -- so some
    ↪ think -- but a good one .
 , ungodly , god - like man , Captain Ahab ; doesn ' t
    ↪ speak much ; but , when h
ll listen . Mark ye , be forewarned ; Ahab ' s above the
    ↪  common ; Ahab ' s been
ewarned ; Ahab ' s above the common ; Ahab ' s been in
    ↪ colleges , as well as ' m
```

These results appear as broken English because the corpus is processed to remove "stop words," i.e. "the, a, that," etc. Run your queries against raw corpus text if you want results in complete sentences.

From the results we can see that he is a Captain of a ship, educated, is "above the common," and keeps his thoughts close to his chest overall. Immediately we find that we can ask Ahab about his ship, his education, his mission (if he'll tell us), etc. If we observe closely we find clues to his personality—the phrase

```
 , ungodly , god - like man , Captain Ahab ; doesn ' t
    ↪ speak much ; but , when h
```

Is intriguing. Let's learn more

```
 >>> text1.concordance('speak')
' t speak much ; but , when he does speak , then you
    ↪ may well listen . Mark ye
```

According to at least one character in the book (who knows Capatain Ahab well as other queries attest), Captain Ahab is a quiet man but should be listened to when he does speak.

Notice through shaping our character so far we've not had to "think" of anything—the system digs through the corpus and provides us with a profile for our subject of interest.

SAVING MIMESIS

Alright, so we write our character based on Captain Ahab. Let's call him, "Captain Moby" (sacrilege, I know):

> Captain Moby leans slightly forward as he scans the horizon from the fore deck of his ship, "*Pequod.*"

An interaction like this:

```
> examine Pequod
I don't see any Pequod here.
```

Roger Sorolla, "Crimes Against Mimesis" (PDF)

would break mimesis. NLTK can help with this by taking our prose through a parts–of–speech parser:

```
>>> nltk.pos_tag(mytext)
[('Captain', 'NNP'), ('Moby', 'NNP'), ('leans', 'VBZ'),
    ↪ ('slightly', 'RB'), ('forward', 'RB'), ('as', 'IN
    ↪ '), ('he', 'PRP'), ('scans', 'VBZ'), ('the', 'DT
    ↪ '), ('horizon', 'NN'), ('from', 'IN'), ('the', '
    ↪ DT'), ('fore', 'NN'), ('deck', 'NN'), ('of', 'IN
    ↪ '), ('his', 'PRP\$'), ('ship', 'NN'), (',', ','),
    ↪  ('Pequod', 'NNP'), ('.', '.')]
```

Here we're looking for Proper nouns (NNP), and nouns (NN). In seconds we're provided the following actionable items:

- Captain
- Moby
- horizon
- fore
- deck
- ship
- Pequod

Does the system respond reasonably to, *"examine Pequod"* or *"examine fore"*? Using the parts–of–speech tager, yes, we're able to cover all relevant areas in each passages' prose. The parser may then be easily written into a script that takes your work's manuscript as input and produces a list of objects for implementation, even, perhaps, creating object templates having "TODO" placed in them to save you the trouble of having to write it yourself.

If we run a NLTK 'similar' query like we did above on the word, *poquod*, by the way, we get this:

```
>>> text1.concordance('pequod',lines=10)
Displaying 10 of 25 matches:
...
rare old craft as this same rare old Pequod . She was a
    ↪ ship of the old school ,
to me and Captain Bildad to see the Pequod fitted out
    ↪ for the voyage , and supp
aling I must , and I would ; and the Pequod was as good
    ↪ a ship as any -- I thoug
...
```

We now see that the *Poquod* is a commissioned ship of the stoutest breed of "old schol" nautical design and construction. So it goes.

Notice overall that the single character in our story is modeled from a single corpus. You can craft your character with references from several bodies of work. Still other characters may be modeled after a collection of completely separate works; the antagonist in your story could be modeled after Shakespeare's *Othello*. There exists virtualy no end to the ways by which you can form custom corpus' to suit the needs of your work of IF.

INSTALLATION

Fortunately, installing NLTK is fairly straightforward; you need only python, NLTK, and NLTK-DATA installed to use the examples described here.

INSTALLING PYTHON

Linux systems, almost without exception *will* have PYTHON and NLTK packages available as standard. Here are the commands to install PYTHON for various distributions:

For Debian/Ubuntu:

```
sudo apt-get install python3 python3-pip
```

For Arch Linux:

```
sudo pacman -S python python-pip
```

Python for windows download page

Python documentation, "Python on Windows"

Windows systems may download PYTHON from the link found in the margin notes on this page. I personally recommend using POWERSHELL when working with PYTHON and NLTK. For industrial grade text processing in environments where Windows must be used as the base operating system it is best to run these application inside a virtual machine running LINUX or BSD. This is as the Windows architecture is not really suited for massive automation.

INSTALLING NLTK

It is almost always better to stay inside the distribution's method for installing PYTHON applications without resorting to PIP or other PYTHON installation systems. This is because conflicts will occur if a dependency is installed with PIP and that dependency is needed by the package manager only to find the dependency is already installed. For NLTK the installation looks like this:

For Debian/Ubuntu:

```
sudo apt-get install python-nltk
```

For Arch Linux:

```
sudo pacman -S python-nltk nltk-data
```

For all operating systems I recommend downloading all the available NLTK packages; you will save yourself a lot of time later not having to worry about whether or not a required package is installed for a given NLTK operation:

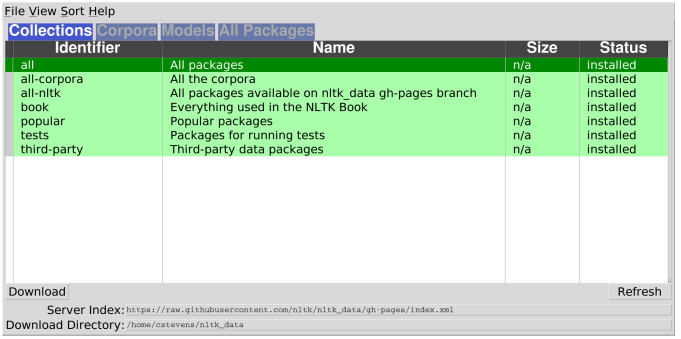
```
>>> import nltk
>>> nltk.download()
```

RUNNING QUERIES

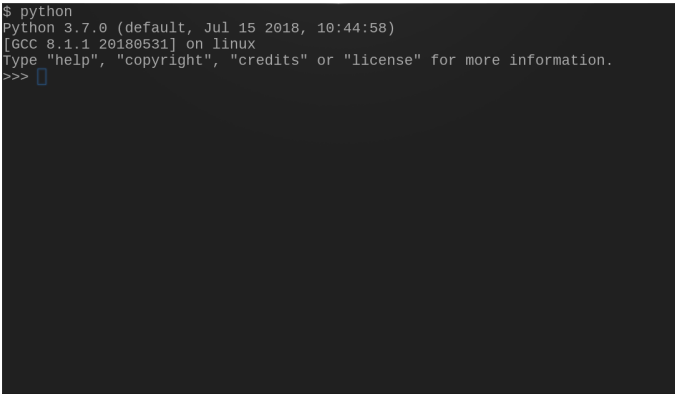
Once PYTHON and NLTK are installed you my now run queries like those outlined at the beginning of the article.

First, enter the PYTHON environment:

```
\$ python
```



NLTK DIALOG FOR installing NLTK data packages



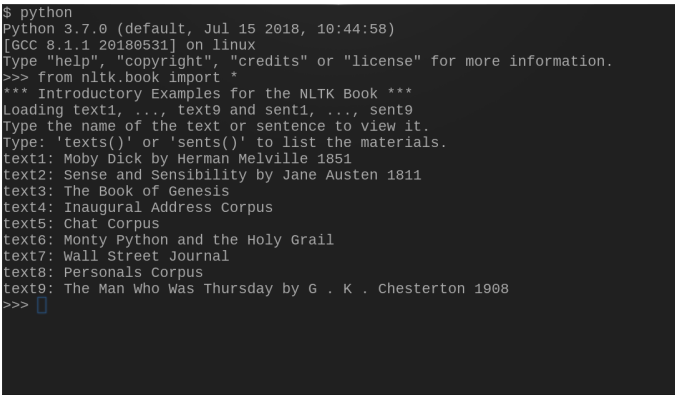
ENTERING THE python environment

Next, we'll load the NLTK library and default text corpus' found included in NLTK:

```
>>> from nltk.book import *
```

Notice the `text1:`, `text2:`, etc. designations. We use these designations to specify a corpus we'd like to use when querying the data:

```
>>> text1.concordance('ahab',lines=2)
```



NLTK AND TEST DATA loaded and ready to receive natural language processing requests

QUERYING THE 'MOBY DICK' corpus with the word, 'ahab'

Naturally, NLTK lets you query differing types of source data for different purposes. Perhaps one character in your narrative is a romantic; you would likely want to get insight into his personality by querying a work like Shakespeare's /textitHamlet. Alternatively, a powerful general's character may be best served by *Secret of the Night* by Gaston Leroux. Differing characters may be formed by pulling from differing

literary works; also worth noting is that several works of the same type may be combined into a single corpora for an even broader scope of natural language processing responses.

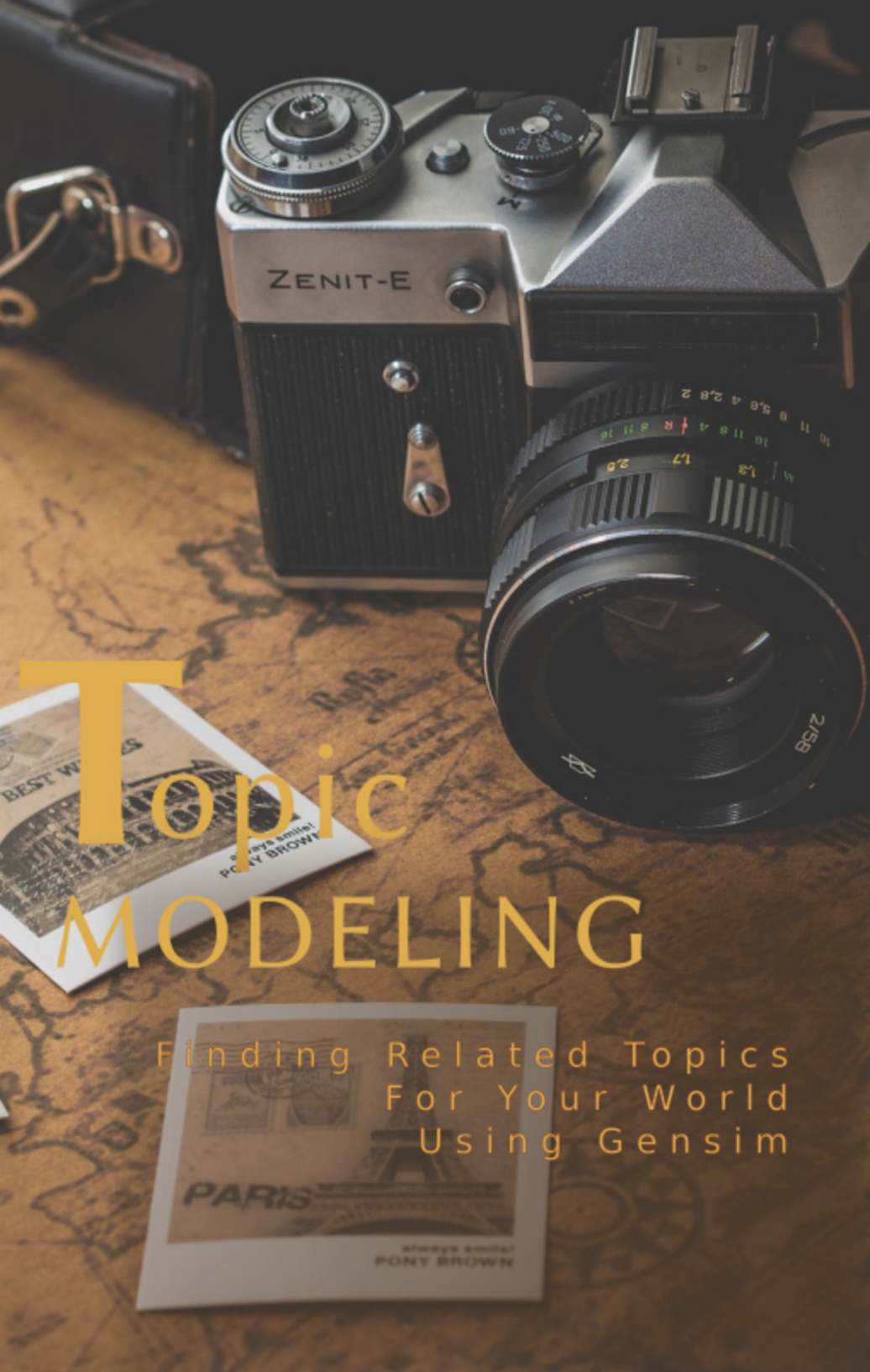
DIVIDE AND CONQUER

Given these tools coupled with custom corpus' it is at least conceivable that scripts could be written to search, tag and template hundreds or thousands of connections automatically for author review and final implementation. Automating NLTK serves itself well to dividing the work into small teams.

For example, after the creators provide the technical staff with the body of works they wish to reference in their story a natural language processing based workflow may look something like this:

- Technologists code & test the automation routines to place NLTK query results into graphs (e.g. GRAPHIZ)
- Authors review the graphs containing the results and entity relationships for quality
- Technologists run automation routines to convert the modified graphs into TADS3 templates
- Artists & authors review the output and modify
- Beta testers test—their transcripts are re–into the natural processing loop as required

All the work, of course, is channeled through a revision system to ensure all affected parties receive timely updates.



Topic MODELING

Finding Related Topics For Your World Using Gensim

III

Topic Modeling

Finding Related Topics For Your World Using Gensim

TOPIC MODELING LIGHTENS the burden of creating complete works of IF

"Writing is an exploration. You start from nothing and learn as you go."

–E. L. Doctorow

Shivam Bansal, Beginners Guide to Topic Modeling Modeling in Python

We now explore the idea of gaining a foothold on our project's topic relationships using gensim. Once you've determined what your work of if is about it would be nice if we had a way of discovering what kind related areas we can find and how all these topics tie together.

RELATIONSHIPS TO DIAGRAMS

When we're finished we'll be able to make queries to the entire work of Wikipedia about virtually any topic imaginable. Here's what a query for *Temple of Athena Nike* looks like:

```
\$ python run_search.py
Loading Wikipedia LSI index (15-30sec.)...
  Loading LSI vectors took 2.53 seconds

Loading Wikipedia article titles...

Searching for articles similar to
'Temple of Athena Nike':
   Similarity search took 1434 ms
   Sorting took 4.40 seconds

Results:
   Temple of Athena Nike
   Temple of Poseidon at Sounion
   Temple of Hera, Olympia
   Temple of Artemis, Corfu
   Athena Alea
   Temple of Olympian Zeus, Athens
   Temple F (Selinus)
   Bassae
   ...
```

Notice that the topic modeller returns a list of related topics by order of relevance. The topic modeler is not a search engine—it's core purpose is not to pull up items *matching* your inquiry, rather, it's purpose is to return items *closely related* to your inquiry. By providing lists of closely related topics to your inquiry you can quickly map the results to items of exploration in your work of if.

Base topic, 'Platonic Realism'

Let's take another example. First, I start with a known article in Wikipedia by looking it up manually. Since topic modelling is not

limited to physical constructs I'd like this time to quickly see what areas are most related to Plato's idea of realism:

```
...
Searching for articles similar to
'Platonic realism':
...
Results:
   Platonic realism
   Ontology
   Indeterminacy (philosophy)
   Theory of forms
   A priori and a posteriori
   Schema (Kant)
   Substance theory
   Nominalism
   ...
```

So from here I can choose Ontology (the study of being) as a preamble to my adventure. I would set the thing up so that the interlocutor must prove that they understand Ontology before moving to the next phase of adventure. Once they do the room lights up to a clear being manifest physically in the updated room description (brightly lit)

I point a sumarizor to the *Platonic realism* topic:

```
sumy lex-rank --length=5 --url https://en.wikipedia.org/
    ↪ wiki/Platonic_realism
```

> In Platonic realism, forms are related to particulars (instances of objects and properties) in that a particular is regarded as a copy of its form.

I point a sumarizor to the Ontology topic:

```
sumy lex-rank --length=5 --url https://en.wikipedia.org/
    ↪ wiki/ontology
```

> This view allows philosophical entities other than actual entities to really exist, but not as fundamentally and primarily factual or causally efficacious; they have existence as abstractions, with reality only derived from their reference to actual entities.


Remember when changing a lightbulb was a hardware problem?
INFOCOM REMEMBERS

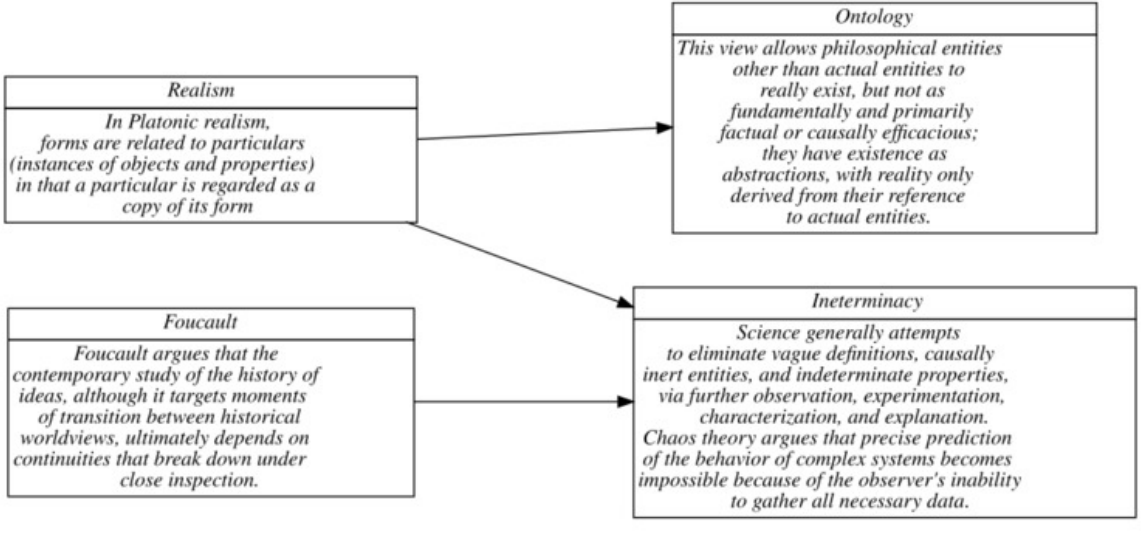
And finally to the second most relevant result, *Interdeterminancy_(philosophy)* topic just for fun:

```
sumy lex-rank --length=5 --url 'https://en.wikipedia.org
    ↪ /wiki/Indeterminacy_(philosophy)'
```

> Science generally attempts to eliminate vague definitions, causally inert entities, and indeterminate properties, via further observation, experimentation, characterization, and explanation. Chaos theory argues that precise prediction of the behavior of complex systems becomes impossible because of the observer's inability to gather all necessary data.

We can now graph the results (I see that the Inteterminancy topic mentions 'Foucault', the author of, "The Archeology of Knowlege," so we'll throw in what he has to say about Interdeterminancy too) using the Graphiz language. In a few minutes and with the source file we get this relationship graph:

The Graphiz graphing file relationships.dot *is attached to this document (if your reader supports it) and is also listed in the end notes.*

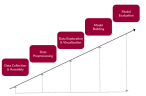


ENTITY RELATIONSHIP GRAPH graph drawn using the results of topic modelling.

TOPIC MODELING PREPARATION

Creating the topic modelling system involves the following:

1. Downloading a complete record of all Wikipedia articles (i.e. a "dump")
2. Converting articles to a "big bag of words"
3. Learning "term-frequency–inverse document frequency" [TF-IDF] from bag of words
4. Applying TF-IDF model to all vectors
5. Formulating a Latent Semantic Index [LSI] via shallow artificial neural network
6. Applying LSI to all vectors



Matthew Mayo, "A General Approach to Preprocessing Text Data"

Fortunately, a script is available that performs all these steps for you. Once you've installed the necessary requirements to your system (outlined below) you'll be ready to run the topic modeling queries as shown beginning on page 26. For effective topic modelling you needn't know the details of the process.

INSTALLING PREREQUISITES

GENSIM requires PYTHON, NUMPY, SCIPY, SIX, AND SMART_OPEN to run. Fortunately, either your operating system has a pre–defined package to install that will install all these dependancies for you or PYTHON's internal PIP installation system will install the required dependencies for you.

Installing from your operating system's package manager is the preferred route as it will avoid conflicts down the road. On Arch Linux, for example, the command to install Gensim through YAOURT is:

GENSIM *quick install guide*

```
yaourt gensim
```

If your system does not have a defined package for GENSIM simply install PYTHON and PIP. From there, install GENSIM using PIP.

PYTHON & PIP *Windows installation guide*

If you have to install GENSIM through PIP (as is slightly prefered in my opionion as PIP generally provides recent versions) then GENSIM may be installed simply by executing:

```
pip install --upgrade gensim
```

GOOD CITIZENRY: DOWNLOADING WIKIPEDIA

Now that GENSIM and it's pre–requisites are installed we can now move to using these tools to download & prepare the Wikipedia article dump for use.

The Wikipedia article dump is big, taking around 15GB and growing. The preferred way to download the dump is through a torrent file. Torrent downloads reduce the load on a server by spreading the bandwidth use across multiple servers. The second preferred way is to download the dump from a Wikipedia mirror. The least preferred way is to download the dump from Wikipedia's server proper.

Wikipedia dump file names are in the form:

```
enwiki-[latest]-pages-articles.xml
```

Download the latest torrent file (enwiki-20170820-pages-articles.xml.bz2 as of this writing) to your normal download directory.

RTORRENT *installation and use guide*

Fire up your torrent client (or double-click the torrent file you just downloaded) and initiate the torrent download process. I personally used the terminal–based RTORRENT but any torrent client will do. Be patient as the Wikipedia dump is a large download and will take time to finish, even if your system is connected to fiber–optic lines.

DOWNLOADING FROM A MIRROR

Wikipedia Mirror List

Alternatively, you can download the dump from a mirror. These are generally FTP sites. The key to finding the file you want to download is to navagate to the *enwiki* subdirectory. An example URL looks like this:

```
http://ftp.acc.umu.se/mirror/wikimedia.org/dumps/enwiki
    ↪ /20180901/enwiki-20180901-pages-articles.xml.bz2
```

Once you've selected the appropriate file simply click the link and download as you normally would.

INSTALLING WIKI–SIM–SEARCH

WIKI-SIM-SEARCH is a python script that automatically processes the Wikimedia articles into forms usable for topic modeling with GENSIM. To install WIKI-SIM-SEARCH you can either download the project's repository using GIT or download the zip file.

WIKI–SIM –SEARCH *Github repository, including instructions for use*

Once WIKI-SIM-SEARCH is cloned or decompressed, copy (or move) the `enwiki-[latest]-pages-articles.xml.bz2` to the `data` directory of the project.

Once the Wikipedia articles are in place simply run:

```
python make_wikicorpus.py
```

to process the Wikipedia dump. This process will take roughly 12 hours on an INTEL I7. The process (at least on LINUX) will seem to not be doing anything; be patient...if you see the cursor just blinking but the program hasn't returned an error it is running and should finish. The output will look similiar to this:

```
Parsing Wikipedia to build Dictionary...
   Building dictionary took 3:05
   8746676 unique tokens before pruning.

Converting to bag of words...
   Conversion to bag-of-words took 3:47

Learning tf-idf model from data...
   Building tf-idf model took 0:47

Applying tf-idf model to all vectors...
   Applying tf-idf model took 1:40

Learning LSI model from the tf-idf vectors...
   Building LSI model took 2:07

Applying LSI model to all vectors...
   Applying LSI model took 2:00
```

You may get a print statement error immediately after running MAKE_WIKICORPUS.PY like this:

Changed print statement in Python 3.0

```
  File "make_wikicorpus.py", line 80
    print 'Parsing Wikipedia to build Dictionary...'
SyntaxError: invalid syntax
```

This is because you are running the latest version of python. Go into the `make_wikicorpus.py` file and add parenthesis to every print statement. For example, change the print statement from this:

```
print 'Parsing Wikipedia to build Dictionary...'
```

To this:

```
print ('Parsing Wikipedia to build Dictionary...')
```

Topic Modeling with GENSIM *affords several ways to find and analyise topics and key word relationships*

Adding parentheses to print statements also applies to the other search functions: SIMSEARCH.PY, KEYSEARCH.PY, and SEARCHWITHWIMWEARCH.PY.

Once the MAKE_WIKICORPUS.PY function finishes you'll be ready to run topic modeling as described at the beginning of this article. To search for a specific topic, change the QUERY_TITLE variable in line 29 of the RUN_SEARCH.PY file to the topic you like.

For example, change line 29 from this:

```
query_title = 'Topic model'
```

To this:

```
query_title = 'Platonic realism'
```

The "trick" is to search for Wikipedia articles titles proper using case sensitive queries. To run the search again on one of the results simply copy & paste the result and modify RUN_SEARCH.PY's `query_title` variable.

Here you have it—a method for quickly pulling in topic relationships for whatever your project calls for. By using Wikipedia as a base you are pulling from a large "real—world" body of information. Even if you simply model your topics loosely (perhaps "Josh Wayne" in your story is closely modeled after "John Wayne") against the Wikipedia corpus your stories will "feel" authentic because they are based on authentic sources.

IV

Notes

Chapter 1 New Horizons

Artificial Intelligence For Engaging World Models

[Page 5 World Champion Gary Kasparov] Gary Kasparov image Copyright 2007, S.M.S.I., Inc. – Owen Williams, The Kasparov Agency, CC BY-SA 3.0

Chapter 3 Topic Modeling

Finding Related Topics For Your World Using Gensim

[Page 29, Graphiz relationship graph code listing]

```
// To run w/ SVG output: dot -Tsvg -o [output file] [
    ↪ input file]
// Example: dot -Tsvg -o relation_graph.svg
    ↪ relationships.dot
digraph {
   graph [pad="0.5", nodesep="0.5", ranksep="2"];
   node [shape=plain]
   rankdir=LR;

Realism [label=<
<table border="0" cellborder="1" cellspacing="0">
<tr><td><i>Realism</i></td></tr>
  <tr><td><i>In Platonic realism,<br/>
forms are related to particulars<br/>
 (instances of objects and properties)<br/>
```

```
 in that a particular is regarded as a<br/>
 copy of its form</i></td></tr>
</table>>];

Ontology [label=<
<table border="0" cellborder="1" cellspacing="0">
<tr><td><i>Ontology</i></td></tr>
<tr><td><i>This view allows philosophical entities<br/>
other than actual entities to<br/>
really exist, but not as <br/>
fundamentally and primarily<br/>
factual or causally efficacious;<br/>
they have existence as<br/>
abstractions, with reality only<br/>
derived from their reference<br/>
 to actual entities.</i></td></tr>
</table>>];

Indeterminacy [label=<
<table border="0" cellborder="1" cellspacing="0">
<tr><td><i>Ineterminacy</i></td></tr>
<tr><td><i>Science generally attempts<br/>
 to eliminate vague definitions, causally <br/>
inert entities, and indeterminate properties, <br/>
via further observation, experimentation, <br/>
characterization, and explanation.<br/>
Chaos theory argues that precise prediction <br/>
of the behavior of complex systems becomes<br/>
impossible because of the observer's inability<br/>
to gather all necessary data.
</i></td></tr>
</table>>];

Foucault[label=<
<table border="0" cellborder="1" cellspacing="0">
<tr><td><i>Foucault</i></td></tr>
<tr><td><i>Foucault argues that the <br/>
contemporary study of the history of<br/>
 ideas, although it targets moments<br/>
 of transition between historical<br/>
 worldviews, ultimately depends on<br/>
 continuities that break down under <br/>
close inspection.
</i></td></tr>
</table>>];


Realism -> Ontology;
```

```
Realism -> Indeterminacy;
Foucault -> Indeterminacy;
}
```