**Wikipedia Question Generator Documentation**

**Purpose and Implementation**

The Wikipedia Question Generator is designed to read a Wikipedia page and convert sentences found there into a question and answer format. For instance, if a Wikipedia page contains the sentence, “George Washington was the first president of the United States of America and the leader of the Colonial Army in the Revolutionary War.”, the Question Generator would convert that sentence into the following questions (and answers):

* Who was George Washington? Answer: the first president of the United States of America.
* Who was George Washington? Answer: the leader of the Colonial Army in the Revolutionary War.
* Who was the first president of the United States of America? Answer: George Washington.
* Who was the leader of the Colonial Army in the Revolutionary War? Answer: George Washington.

The usefulness of these questions for the purposes of creating a quiz is currently subject to evaluation by a person. For instance, “Who was George Washington?” is asked multiple times and has multiple answers, suggesting it may be too open-ended for a quiz question. The last two questions, however, seem quite suitable for a short answer quiz format. Defining what makes for good and bad questions in this fashion can inform a rule-building process that can be encoded into the question generator over time. Rules can even be customized on a user-by-user basis depending on what they feel makes for a good question and a bad question.

The current version of the Question Generator is intended to successfully convert only very simple sentence formats, but ones that are commonplace on Wikipedia. Additionally, the Question Generator is intended to provide output that can be used to train next versions of the tool to correctly convert more sophisticated sentences and topics. When an attempt to generate questions is unsuccessful, the output indicating how noun and verb phrases were parsed, how words were POS-tagged, and what entity recognition took place, this information can be incorporated by a programmer to add new regular expressions and programming logic to address new complexities and make this Question Generator into an increasingly sophisticated tool.

**Instructions for Use**

*Input*

The tool requires a simple input. Assuming one knows how to run python within their operating system, one uses the appropriate command line interface to run the program with the term that they wish to find in Wikipedia and convert into questions. The following instructions work for the Windows command line (assuming that python exists as part of the path environmental variable in Windows).

1. Need to install two libraries if they are not already installed into your python environment:

* pip install Wikipedia
* pip install nltk

Note that some python commands download specific pieces for using nltk and that these may be commented out after initial instillation.

1. Within the command line interface, navigate to the folder that contains the python code “questionizer.py”.
2. Type “python questionizer.py [topic]

[topic] can be any single or multi-word term. (Examples: Winston Churchill; Boston; The Silence of the Lambs)

*Output*

Output for the program is currently extensive as this allows for some analysis for the purposes of program refinement. The following outputs are produced:

* Topic:

The title of the Wikipedia page that was selected based on the term entered.

* Summary Paragraph:

The summary paragraph from the selected Wikipedia page is printed tokenized into sentences.

* Sentence to be Converted (tokenized):

The first sentence from the summary paragraph is printed as word-level tokens.

* POS Tagged Sentence:

The same sentence is reprinted with POS tags applied to each of the words.

* Sentence with entity recognition:

A word tree of the sentence with entity recognition applied.

* POS Tagged and Chunked Sentence as Word Tree:

A word tree of the sentence as broken into identified noun and verb phrases. No entity recognition is provided in this tree.

* Noun and Verb Phrases to be used in questions and answers:

Based on the chunked sentences, the specifically identified noun and verb phrases are printed on single lines for each identified phrase.

* Sentence Suitability:

Based on the expectation of the current question generator to produce suitable questions, this provides a guess as to how well the generator will perform. If the sentence was of a simple, suitable format, it will predict that the sentence appears to fit the question creation format. It not, it will predict that the sentence may not be suitable.

* Entity Type for determining question word:

Prints the entity type of the word that will be used to provide the question word. For instance, a “PERSON” should result in the question word “Who”. The simple system for determining this at this time is to take the very first entity type found within the sentence.

* Question word:

The word that will lead the question as determined by a function that returns appropriate words based on the entity type fed into the function.

* Generated Questions and Answers:

A list of all questions and answers generated by the question generator. No suitability analysis is conducted. It simply combines the noun and verb phrases to produce possible questions and answers to be evaluated by the operator at this time. This will allow for rule creation for an eventual evaluation component.

While the questions and answers are the ultimate purpose of this tool, all of this output is provided for the purposes of evaluating effectiveness of the tool in its current state to refine it for its eventual full operationalization.

**Some use cases and expected results**

For demonstration purposes, a few use cases are provided. Inputs, expected results (i.e., results when I tried the software) and possible evaluations (that could inform future rule-building) are given.

*Use case 1: George Washington*

* Input: python questionizer.py George Washington
* Generated Questions and Answers:
  + Question: Who was George Washington? Answer: one of the Founding Fathers
  + Question: Who was George Washington? Answer: the first President of the United States
  + Question: Who was George Washington? Answer: 1789–1797
  + Question: Who was one of the Founding Fathers? Answer: George Washington
  + Question: Who was the first President of the United States? Answer: George Washington
  + Question: Who was 1789–1797? Answer: George Washington
* Evaluation:
  + Question Word: Who

This is the correct word. Questions are about a person.

* + Questions and answers:

The first three identical questions, “Who was George Washington?” are appropriate in terms of format but probably not suitable for a quiz if the answers provided are meant to be absolute. Who was George Washington? Is a complicated question that could have many possible correct answers and would likely require at least an essay.

The question, “Who was one of the Founding Fathers?” probably would not work because it has multiple possible correct answers.

The question, “Who was the first President of the United States?” works very well as a question with only one correct answer, which is the answer provided.

The last question is nonsense as a product of the dates being considered a noun phrase.

*Use case 2: Teamsters*

* Input: python questionizer.py Teamsters
* Generated Questions and Answers:
  + Question: What is The International Brotherhood of Teamsters? Answer: a labor union in the United States
  + Question: What is The International Brotherhood of Teamsters? Answer: Canada
  + Question: What is a labor union in the United States? Answer: The International Brotherhood of Teamsters
  + Question: What is Canada? Answer: The International Brotherhood of Teamsters
* Evaluation:
  + Question Word: What

This is appropriate. Questions are about an organization.

* + Questions and answers:

The first two questions, in terms of format, “What is the International Brotherhood of Teamsters?” is a well-formulated question. The answers provided show that there is a need for more sophistication in terms of assembling noun phrases. The first answer works well but is incomplete and should have “and Canada.” at the end. Sometimes “AND” represents a new independent phrase and sometimes it does not. The nuance of this is complex and needs appropriately sophisticated rules.

The third question: “What is a labor union in the United States?” and its answer are reasonable, but don’t work well as it would not be a single correct answer – other unidentified correct answers are possible.

The fourth question is nonsense.

*Use case 3: Humphrey’s Peak*

* Input: python questionizer.py Humphrey’s Peak
* Generated Questions and Answers:
  + Question: Who is Humphreys Peak? Answer: the highest natural point in the U.S. state of Arizona ,
  + Question: Who is Humphreys Peak? Answer: an elevation of 12,637 feet
  + Question: Who is Humphreys Peak? Answer: 3,852 m
  + Question: Who is Humphreys Peak? Answer: the Kachina Peaks Wilderness in the Coconino National Forest ,
  + Question: Who is Humphreys Peak? Answer: 11 miles
  + Question: Who is Humphreys Peak? Answer: 17.7 km
  + Question: Who is Humphreys Peak? Answer: north of Flagstaff , Arizona
  + Question: Who is the highest natural point in the U.S. state of Arizona ,? Answer: Humphreys Peak
  + Question: Who is an elevation of 12,637 feet? Answer: Humphreys Peak
  + Question: Who is 3,852 m? Answer: Humphreys Peak
  + Question: Who is the Kachina Peaks Wilderness in the Coconino National Forest ,? Answer: Humphreys Peak
  + Question: Who is 11 miles? Answer: Humphreys Peak
  + Question: Who is 17.7 km? Answer: Humphreys Peak
  + Question: Who is north of Flagstaff , Arizona? Answer: Humphreys Peak
* Evaluation:
  + Question Word: Who

This should be “What”, in some cases possibly, “Where” and as the generator grows in sophistication, might be able to identify questions like “How tall?”. “Who” was identified because “Humphrey” is a PERSON whereas Humphrey’s Peak is a thing or a location.

* + Questions and answers:

If the first word was corrected to “What”, a couple of these questions work very well:

* “What is Humphreys Peak? Answer: the highest natural point in the US state of Arizona”
* “What is the highest natural point in the US state of Arizona? Answer: Humphrey’s Peak”

If the first word was corrected to “Where”, a couple others work well (and would work better as noun phrase concatenation becomes more sophisticated):

* “Where is Humphreys Peak? Answer: the Kachina Peaks Wilderness …”
* “Where is Humphreys Peak? Answer: north of Flagstaff, Arizona”

There are also some sophisticated possible questions about the altitude of the mountain that seems possible with complex topic identification and evaluation not just of parts of speech but meanings of terms – a realistic future state for a tool such as this, but not a short term one.

Other use cases that can be tried out with similar good and bad questions, sometimes correct or incorrect question words. Some interesting examples (illustrating just the interesting parts):

*Use case 4: Cool Hand Luke*

Cool Hand Luke produced the question word “Who”, which seems appropriate given that the character Cool Hand Luke is, indeed a person. But the Wikipedia page deals with the film itself and thus should be a “What”. The question, “Who is Cool Hand Luke?” has an answer: “a 1967 American prison drama film directed by Stuart Rosenberg.” This would work well if the question word was “What”.

*Use case 5: Fallout 4*

This is a video game. The correct Wikipedia page was identified, but part of speech tagging decided that “Fallout” is a preposition and didn’t include it in the noun phrase. A good question is produced but eliminates a key word:

“What is 4? Answer: a post-apocalyptic action role-playing video game developed by Bethesda Game Studios.”

*Use case 6: Christmas* AND *Use case 7: Christmas Day*

Using the term “Christmas” supplied the Wikipedia page for “Christmas Christmas”, which resulted in an unexpected question and answer: “Question: Who is the nineteenth studio album by American rock band Cheap Trick? Answer: Christmas Christmas”. While tagging “Christmas” as a person, it surprisingly supplied information about a rock music album.

“Christmas Day” supplied the correct page for Christmas and identified it as a “GPE” that supplies the question word, “What”. The noun phrases were too complex to produce useful questions and answers, but one can see that, with more sophisticated concatenation, appropriate questions could be produced:

* Question: What is Christmas? Answer: an annual festival commemorating
* Question: What is Christmas? Answer: the birth of Jesus Christ ,
* Question: What is Christmas? Answer: December 25 as a
* Question: What is Christmas? Answer: cultural celebration among billions of people around the world
* Question: What is an annual festival commemorating? Answer: Christmas
* Question: What is the birth of Jesus Christ ,? Answer: Christmas
* Question: What is December 25 as a? Answer: Christmas
* Question: What is cultural celebration among billions of people around the world? Answer: Christmas

*Overall evaluation*

As expected, the Question Generator is currently very primitive and effective only to the extent that it finds a simple term and is subject to the formatting of a given Wikipedia page. Some good questions can be produced, and output is supplied to identify rules for better question generation and for eventual automated question evaluation.

**Next Steps**

The Question Generator is currently intended to provide a simple demonstration of question conversion and to provide output that can inform further training for conversion of increasingly complex sentences. As such, it does not yet convert a whole page, but demonstrates question conversion based on a single sentence within the document. The looping logic to convert a whole page is not especially complex, but the output would be overwhelming given the current focus on tool improvement, whereas the work done on a single sentence is illustrative of the capabilities and current shortcomings of the tool and is very useful for demonstration and for evaluation.

Next steps for the software are largely dictated by what is learned through its use. Identifying the sentence structures that exist within Wikipedia and building the regular expressions and entity recognition that can address them will take time. As this becomes adequately sophisticated, adapting the tool from a self-training instrument into a useful quiz generator will make more sense. At that time, adding the looping logic to convert entire pages of sentences into questions will also be provided. An important piece when that is implemented is something that can relate sentences so that pronouns are replaced by the terms to which they refer. For a pair of sentences like, “George Washington was the first US president. He commanded the colonial forces in the Revolutionary War,” “He” would be replaced by “George Washington” for the purposes of question and answer creation.

An additional next step will be to add a question evaluator so that provided output will be limited only to questions that are already determined to make sense and be suitable for a quiz.

**References**

Heilman, M. (2011), Automatic Factual Question Generation from Text, Doctoral Dissertation, Carnegie Mellon University, Pittsburg PA

Petschauer, L. (2015), Linguistics for Pythonistas: Why noun phrase chunking with the NLTK is awesome and useful. Presentation at PyCon, Montreal QC

TextMiner (2014), Dive into NLTK, Part III: Part-Of-Speech Tagging and POS Tagger