broccoli

Emilie McConville
Zach Paine
Mattt Thompson
Owen Yamauchi

11-411 Natural Language Processing
Noah A Smith, Spring 2008

Hello, we're team Broccoli
(Like everyone else who couldn't come up with a team name, we got a vegetable, but as far as we're concerned, broccoli is the best)

Approach

(How we thought it should work)

From the start, we had some good ideas of how to approach the task at hand. However, if we were to summarize our plans into one concept, it would be...

NER

Named Entity Recognition
Why?
- Most elements of interest are going to be uncommon words / entities
- Use this fact to generate interesting questions

Also,

- POS Tagging
- Stemmer
- Anaphora / Co-reference Resolution
- Article Classification

We also anticipated other traditional NLP tools to be useful to refine what we were generating with our NER approach

Implementation

(How it actually ended up working)

...Of course, plans are subject to change, and as we went along with the project, we found out what worked and what didn't.

- We couldn’t find a way to use NER for answering.
- Using only sentences with named entities to generate questions didn’t improve the quality of the questions (although it did make it easier to choose the correct wh-word).

Keyword Matching

(For Question Answering)

What we found effective:
For answering questions, we would remove common words (articles, auxiliaries) and collect remaining tokens. We would search through article and find the sentence with the most matches and return it as the answer.
- Used as first (throwaway) approach, but was really effective
- Iteratively added improvements to increase fluency

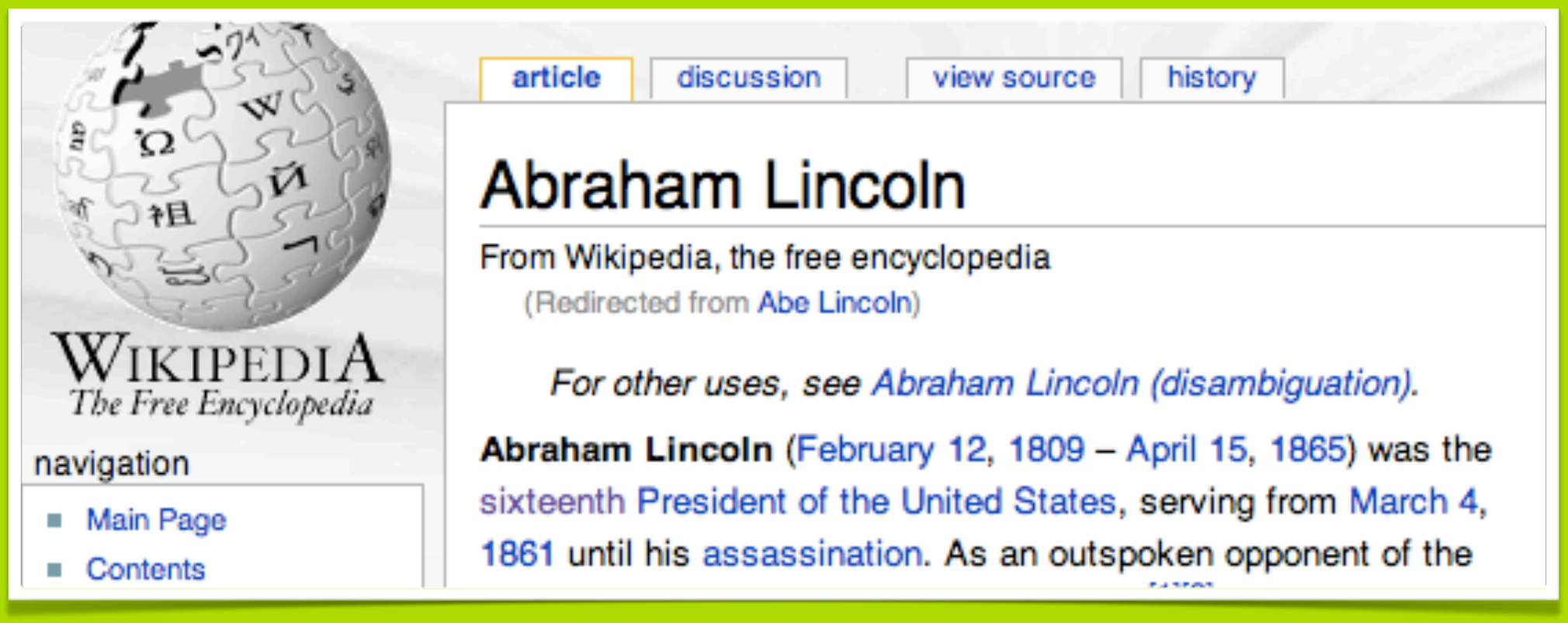
Keyword Matching

Question: *"Who was the sixteenth president of the United States?"*

Keywords: `[sixteenth, president, united, states]`

For answering questions, we would remove common words (articles, auxiliaries) and collect remaining tokens

Keyword Matching



Keywords: `[sixteenth, president, united, states, serving, assassination]`

Score: 4/4 ✓

- We search through article and find the sentence with the most matches and return it as the answer
- If fewer than half of the question's keywords are found, we give up

Syntax Parsing

(For Question Asking)

Using the Stanford Parser, we found sentences in the article that corresponded to sentence structures that we identified to be good indicators of a structure that we could turn into a question.

2 Modules:

FactoidMatcher

“ Who achieved international fame as the leading Union general in the American Civil War?”

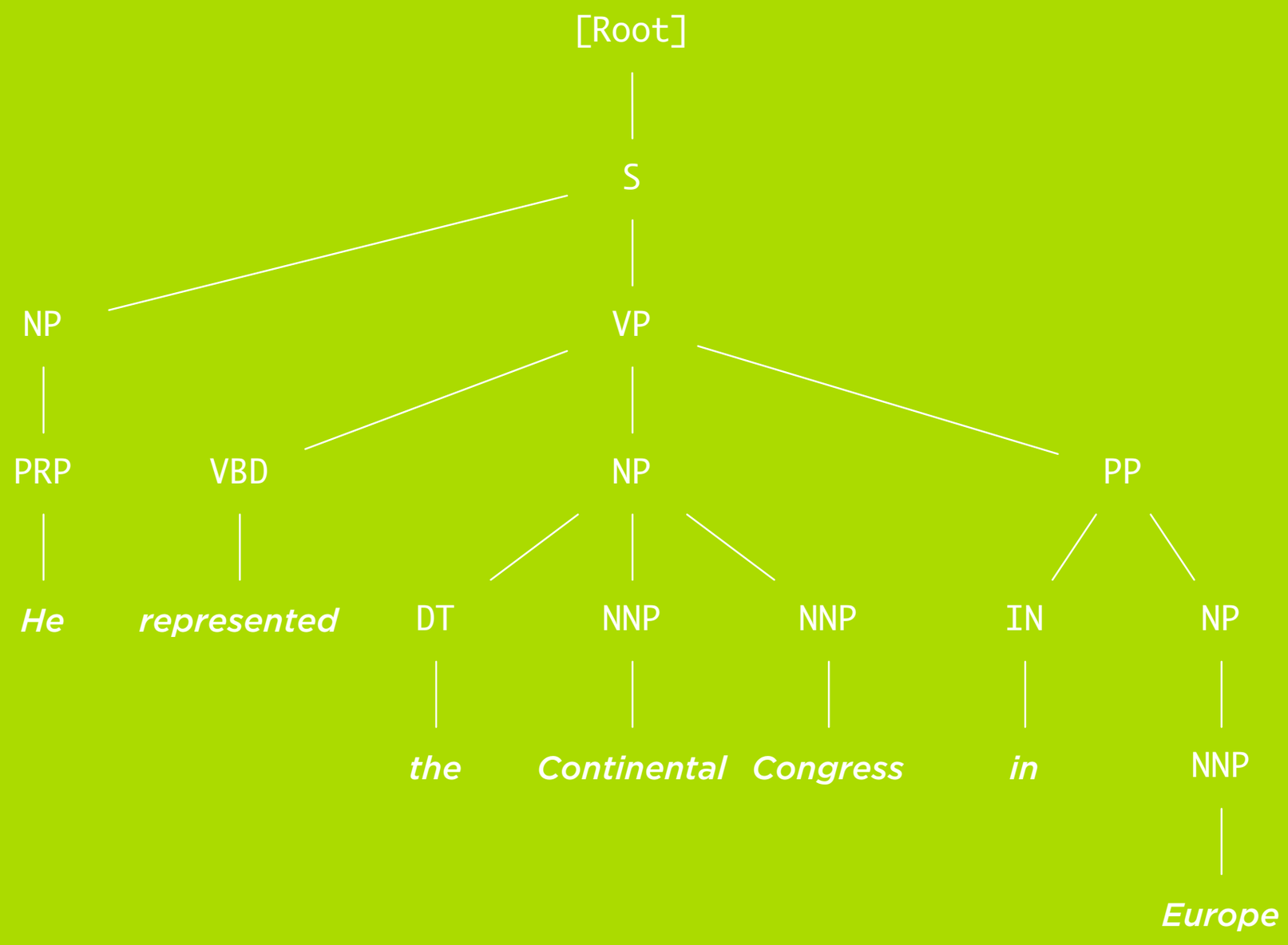
DateMatcher

“ Grant was elected president as a Republican in what year?”

- FactoidMatcher searches for (S NP VP) and replaces the NP with a wh-word.
- DateMatcher searches for (S (PP IN NP) , NP VP), deletes the PP, and adds “in what year?” at the end.

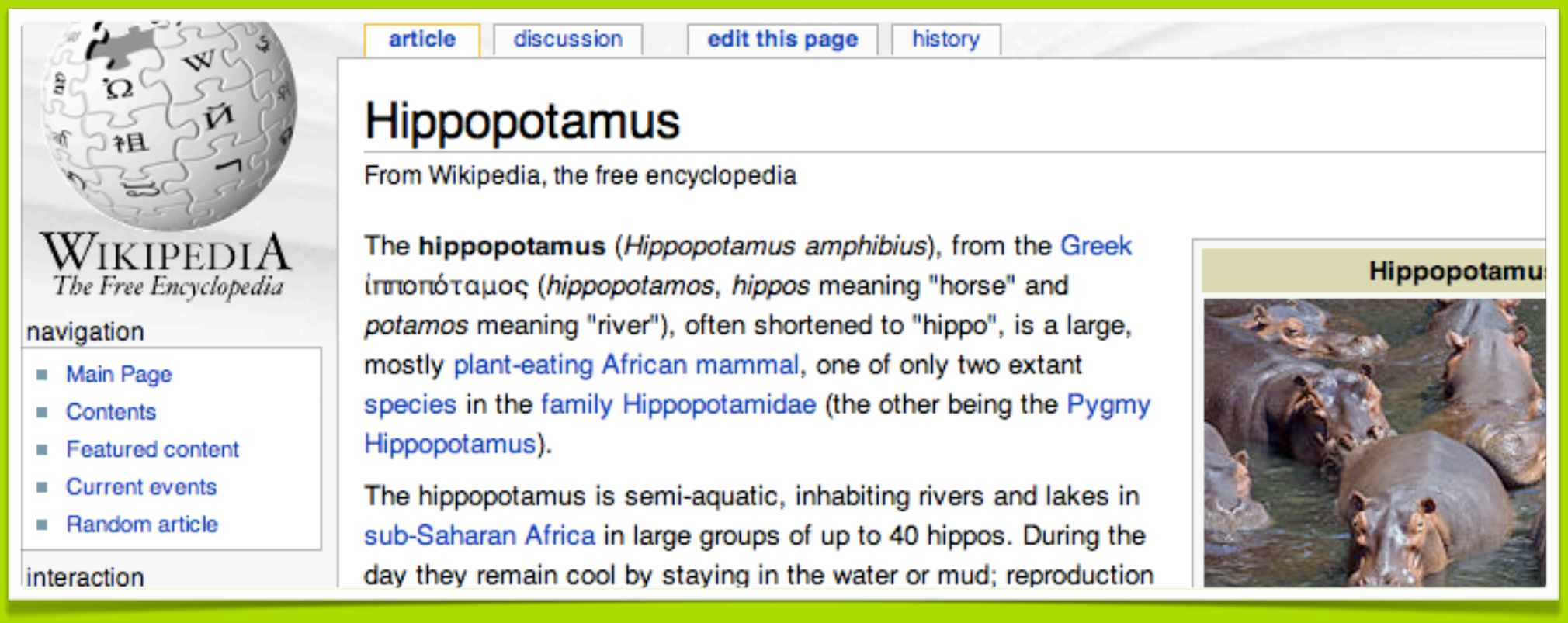
Syntax Parsing

Search With: FactoidMatcher



Search through article to find sentences with (S (NP VP)) structure and replace NP with wh-word

Syntax Parsing



Candidate: *"They emerge at dusk to graze on grass"*

Question Form: `Who emerge at dusk to graze on grass?`

After finding a candidate sentence, we use the POS tagging data to determine the correct question word (ie. who, whose, what)

Other Strategies:

Limit keyword length to [4, 15]

Discard questions where answer was substring of title

Pronoun substitution

- Limit keywords to prevent silly, short questions or really long/ verbose
- Discard questions to avoid answers to all of them being “John Adams”, eg
- Some heuristics to deal with pronouns. If the first word is “it”, the referent must be in a previous sentence; include the previous sentence in the answer. If it’s “he”, “she” or “they”, substitute in the title of the article, pluralizing it if it’s “they”. (Of course the latter heuristic doesn’t always work perfectly, but it’s right quite often.)”

If We Had More Time

Implement Medium & Hard Question Asking

Add more Question Answering Matchers

More levels of Question Answering

- Medium difficulty questions were generated by extracting named entities and adding “what is” or “who is”
- Hard difficulty questions were gnerated based on an article classifier and general-purpose questions that all articles of that type would have (ie. “When was [president] born”, or “What does [animal] eat?”
- More matchers, perhaps Location, eg “Where did something happen”
- QA only works by keyword matching; would be nice to have system that took over when we couldn’t find a match

broccoli

Emilie McConville
Zach Paine
Mattt Thompson
Owen Yamauchi

11-411 Natural Language Processing
Noah A Smith, Spring 2008

Thanks!
Q/A